ABC INFORMATIQUE

N° 43

COURS D'INFORMATIQUE PRATIQUE ET FAMILIALE

Les imprimantes à marguerite

Le Torch Disk Pack

Jeu : "Jacques a dit..."

Conception d'un programme

EDITIONS ATLAS

M6062-43-12F 85FB-3,80FS-$1.95

aBC INFORMATIQUE

cours d'informatique pratique et familiale

EDITIONS ATLAS

Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris *Cedex 15*. Tél. : (37) 28-10-10. Services administratifs et commerciaux : 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Réalisé par EDENA s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris *Cedex 15*.

Direction éditoriale : J.-Fr. Gautier. *Service technique et artistique* : F. Givone et J.-Cl. Bernar. *Iconographie* : J. Pierre. *Correction* : B. Noël.
Publicité : Anne Cayla. Tél. : 202-09-80.

VENTE AU NUMÉRO

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser aux services commerciaux des ÉDITIONS ATLAS, Z.I. de Lucé, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles permettant de relier 12 fascicules sont en vente chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume IV, n° 43.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel *(secrétariat de rédaction)*, Jean-Pierre Bourcier *(coordination)*, Patrick Bazin, Jean-Paul Mourlon, Claire Rémy *(traduction)*, Ghislaine Goullier *(fabrication)*, Marie-Claire Jacquet *(iconographie)*, Claire Bischoff *(correction)*.
Crédit photographique, couverture : F. Merlin-TDF.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : novembre 1984. 98411. Dépôt légal en Belgique : D/84/2783/27.

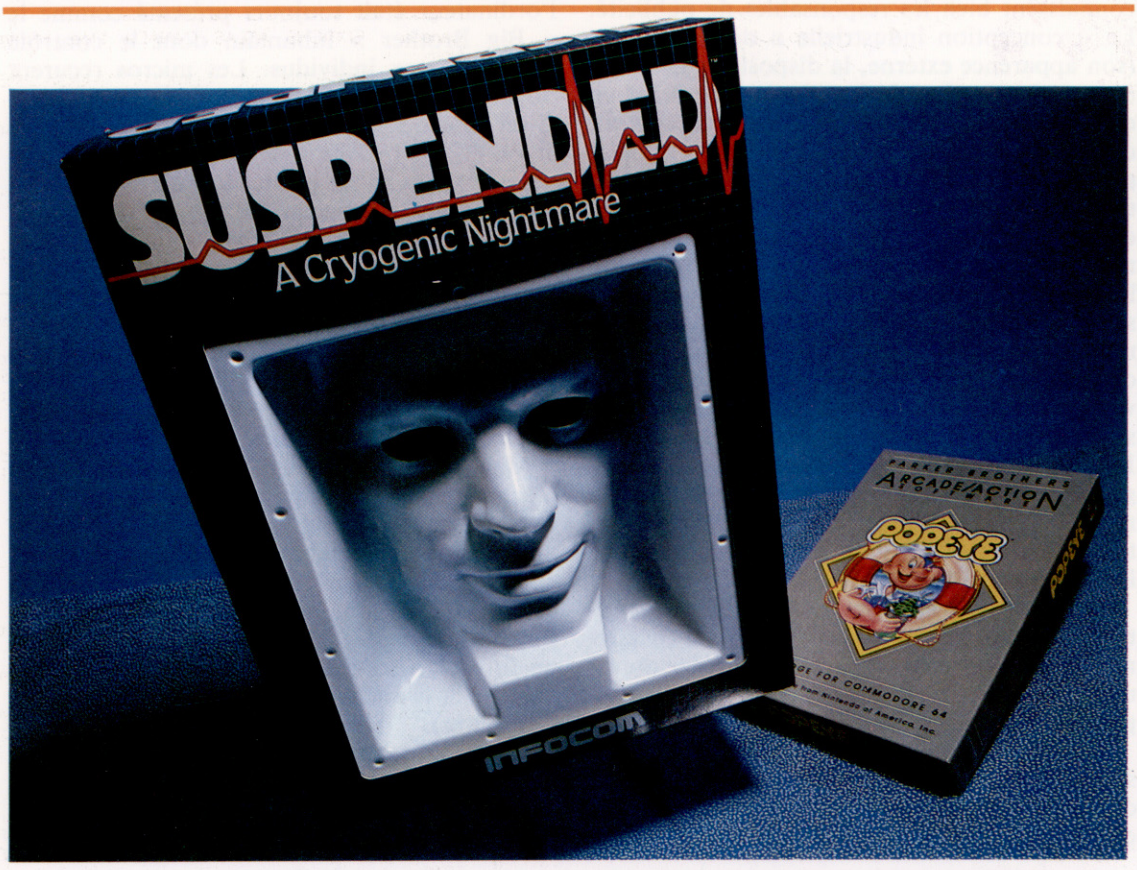
A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas

Marketing d'abord

Pour vendre un programme de jeu ou un nouvel ordinateur de gestion il faut d'abord créer une bonne image de produit. Le marketing a pris le pas sur la qualité technique. A la clé : l'échec ou le succès.



Vous ne pouvez juger un livre...
Ces deux programmes sur disquettes coûtent environ 480 F chacun, mais l'un d'entre eux est emballé d'une façon qui attire l'œil et suggère un bon rapport qualité/prix, tandis que l'autre semble n'être qu'une vulgaire boîte vendue à un prix très élevé.
(Cl. Pilot Software Liz Heaney.)

Le marketing est l'activité qui consiste à construire un pont entre le produit à vendre et le portefeuille des acheteurs éventuels. En construisant un pont qui encouragera et permettra un maximum de circulation, les responsables du marketing doivent pressentir (en se basant souvent sur des études de marché) les besoins, les désirs et les caprices de leur public, puis investir massivement pour répondre à la demande. Par exemple, le récent lancement du Macintosh au plus fort de la campagne coûtait jusqu'à 12 millions de francs par mois, uniquement pour la campagne se déroulant en Grande-Bretagne.

Les premières décisions de marketing sont prises lors du stade de la planification de la production d'une nouvelle machine. Quelles fonctions doit posséder cette machine, combien doit-on construire d'unités, et quel peut être le coût de fabrication pour chaque unité, voilà des facteurs qui vont guider la démarche marketing.

Une réflexion similaire s'applique à la mise sur le marché des logiciels. Il n'est pas recommandé de dépenser des sommes énormes à développer un jeu qui, pour rentabiliser l'investissement, coûtera plus que ne peuvent payer les acheteurs potentiels. Mais une société de logiciel avare en travaux de développement proposera un produit comportant peu de fonctions, ou bien des défauts de mise au point (ou ces deux lacunes à la fois), courant ainsi le risque de compromettre l'image de marque de la société et de nuire à la vente de futurs produits. Si elle dépense de fortes sommes en développement mais néglige tout l'aspect marketing, elle aura peut-être créé un excellent produit, mais personne n'en connaîtra l'existence.

Il est aussi important de déterminer — dès le début — où se situe le produit par rapport aux produits similaires disponibles dans les boutiques. C'est là que l'image du produit prend toute sa signification. Les fabricants de cigarettes et de lessives ont à faire face au même problème — en fait tous ces produits se valent. Les fabricants de matériel informatique ne sont pas vraiment dans cette situation; néanmoins, ils doivent expliquer précisément (et ce n'est pas toujours facile) en quoi leur machine se distingue des autres à des acheteurs débutants qui comprennent difficilement les détails techniques des explications. Les jeux et les programmes de

gestion ne révèlent souvent leurs meilleures caractéristiques qu'après avoir été achetés et utilisés.

C'est pourquoi les fabricants de matériel et de logiciels, comme les fabricants de cigarettes et de lessive, ont recours à une « image produit », sorte de projection psychologique du produit. Pourquoi? Parce qu'il est plus facile de vendre une idée qu'un produit; c'est un principe que connaissent bien les responsables de publicité. La « conception industrielle » de la machine (son apparence externe, la disposition des boutons et des touches de fonction) est souvent un excellent point de départ dans le développement d'une image de produit. Le BBC, par exemple, a mis l'accent sur une apparence simple et utilitaire qui correspond bien à l'image du produit éducatif qu'il veut projeter. Le boîtier d'autres machines comporte souvent des touches moulées plus rapides portant des inscriptions amusantes qui séduisent l'amateur de jeux. Les ordinateurs Atari — dont on a modifié le style au cours d'une campagne de promotion pour sauver le produit en 1983 — ont une apparence sévère et militaire parfaite pour jouer à « Tank Battle ». Commodore, dans son combat pour gagner le marché américain, a demandé à Ferdinand Porsche, le concepteur de la fameuse voiture, de créer pour sa gamme de produits une ligne sobre et futuriste.

A l'opposé, l'apparence physique du Spectrum, avec son petit boîtier noir et ses touches multifonctions complexes et élaborées, donne une image de densité qui suggère un bon rapport qualité/prix.

L'« image de produit » ne s'arrête pas à la simple apparence physique de la machine. Elle doit être promue par une campagne publicitaire de choc qui renforce l'idée que l'on veut faire passer. L'« énorme » mémoire du Commodore 64 est mise en valeur en utilisant un éléphant dans la publicité des journaux et de la télévision. Pour le BBC, le sigle du hibou évoque l'idée de sagesse.

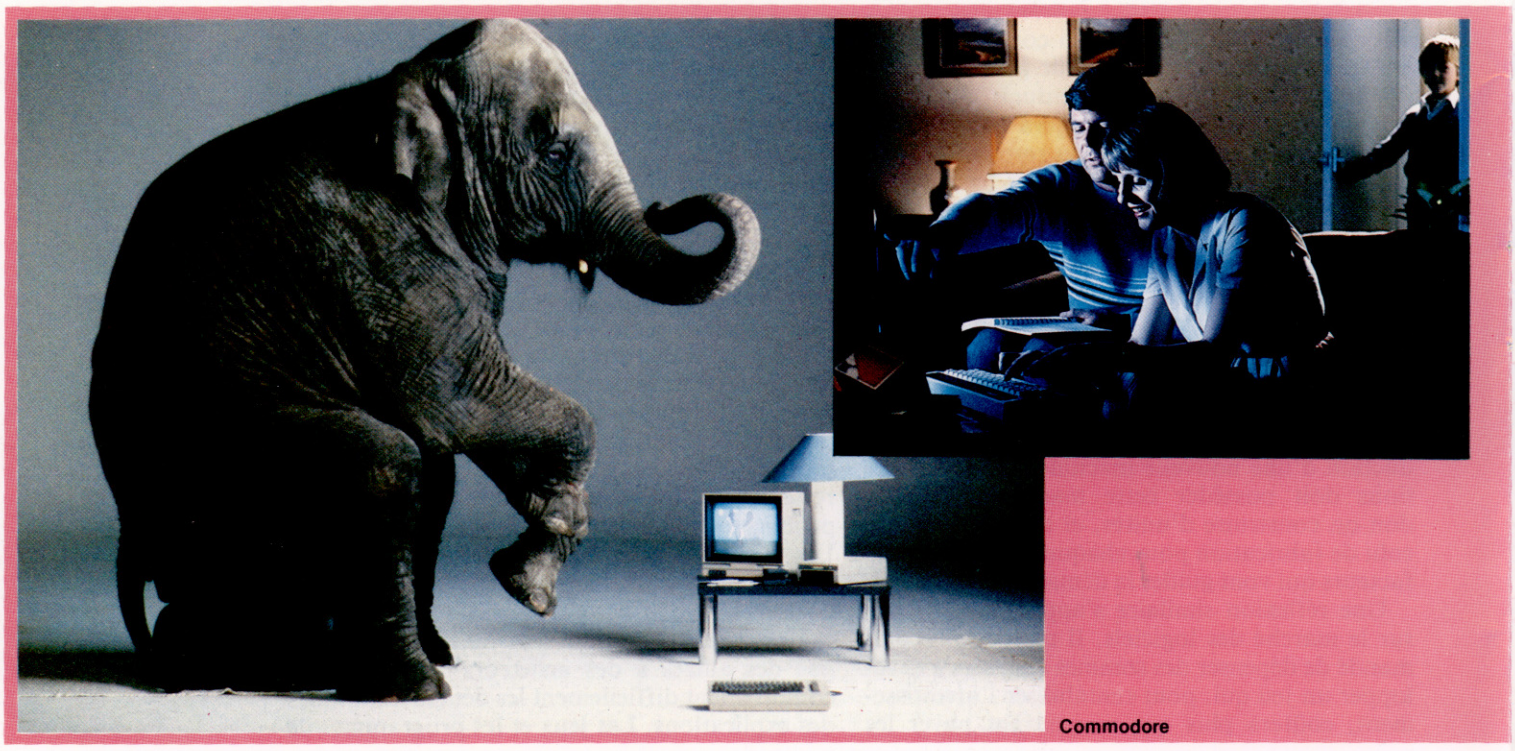
Naturellement, le nom est un élément important. Les premiers ordinateurs domestiques ont eu à lutter contre une image ancrée dans les esprits par les films des années 60 et 70, où l'ordinateur était toujours présenté comme le « Big Brother » inhumain dont le contrôle échappait aux individus. Les micros reçurent donc des noms qui combattaient ce préjugé. C'est ainsi qu'apparurent les ordinateurs Apple et PET.

Le Macintosh fut présenté aux États-Unis au moyen d'une campagne télévisée où une femme démolit l'écran d'un gros ordinateur pour symboliser la nouvelle liberté offerte par le Macintosh. Le slogan utilisé était : « Grâce au Macintosh, 1984 ne sera pas 1984. » Aux États-Unis, Mackintosh désigne une variété de pommes très connue. L'équipe de développement, qui n'était pas très douée pour l'orthographe, oublia le « k » et l'erreur ne fut pas corrigée.

Mais les gens doivent être persuadés qu'ils n'achètent pas un jouet. Commodore eut pour cette raison beaucoup de mal à percer dans le marché professionnel avec le PET, à la fin des années 1970. Ils tentèrent de rassurer les gens en disant que le nom n'était que l'acronyme de Personal Electronic Transactor, mais cet effort ne fut pas assez persuasif et ils décidèrent de renommer le micro CBC (Commodore Business Computer).

La tradition du « hi-tech » recommença à plaire lorsque les mentalités s'habituèrent à la présence de l'ordinateur dans l'environnement quotidien. L'efficacité des noms de produit est

Images
La puissance de l'image dans une campagne de publicité est clairement démontrée dans notre exemple : l'éléphant de Commodore représente l'« énorme » mémoire du Commodore 64; les scènes de la vie domestique et scolaire du BBC laissent entrevoir la souplesse, la facilité d'utilisation et l'importance éducative qui caractérisent le produit, tandis que la publicité d'Apple promet à ses utilisateurs de les introduire dans un monde informatisé plus humain. Comme toute technique moderne, cette approche peut avoir des effets inattendus : l'éléphant craint les souris (contrairement au Lisa et au Macintosh), et peut laisser entendre que le produit est démodé; les acheteurs potentiels de l'Electron peuvent trouver la scène familiale ridicule et stéréotypée, et le lien scolaire intimidant; les clients d'Apple peuvent penser qu'on les présente comme des individus peu intelligents.



Commodore

fonction de la distance créée entre eux et les mots usuels et familiers. Ainsi, des acronymes ou même des combinaisons de lettres sans signification sont préférés à des mots présents dans le dictionnaire. Les lettres rares, qui donnent de très bons résultats de Scrabble, sont également très rentables. D'où des machines nommées ZX81, TO7, MZ-700.

L'emballage, qui n'a qu'un rôle protecteur dans le cas du matériel, a en revanche une très grande importance pour l'image du logiciel. Dans ce dernier domaine, deux stratégies sont offertes aux vendeurs : ne dépenser qu'une somme minimum pour l'emballage (la boîte de la cassette avec un petit dépliant couleur) et offrir le produit avec rabais, ou améliorer l'image du produit en l'offrant dans une grande boîte, ressemblant souvent à un livre, en ajoutant des gadgets supplémentaires qui peuvent ou non faire partie du jeu. Le « Hobbit », par exemple, est livré avec l'édition de poche du fameux livre de Tolkien.

L'image est amplifiée par la publicité. Il existe la même relation entre le marketing et la publicité qu'entre la stratégie et la tactique. Le moment de lancer la publicité, la taille de la campagne sont des décisions relevant du marketing. Un fabricant d'ordinateur domestique lancera un nouveau produit peu avant Noël et investira une partie importante du budget annuel de publicité lors de ce lancement.

Mais la promotion du produit n'est pas toujours utile lorsque l'image est mauvaise. Une campagne de publicité pour une marque de cigarettes s'est rendue célèbre chez les publicitaires. Cette marque fut largement présentée dans des spots publicitaires à la télévision et au cinéma avec une scène montrant un homme seul, vêtu d'un imperméable, avec le commentaire : « Vous n'êtes jamais seul avec une ... ». Mais le message qui passa fut celui-ci : « Les solitaires fument des ... », et la marque disparut.

Certaines images proposées pour des ordinateurs peuvent avoir ce genre d'effet négatif. L'éléphant du Commodore peut rappeler que le 64 convient mal à l'écriture de programmes. Plusieurs distributeurs essaient de minimiser le caractère éducatif de tel ou tel micro, craignant que cela ne finisse par nuire à la vente du produit.

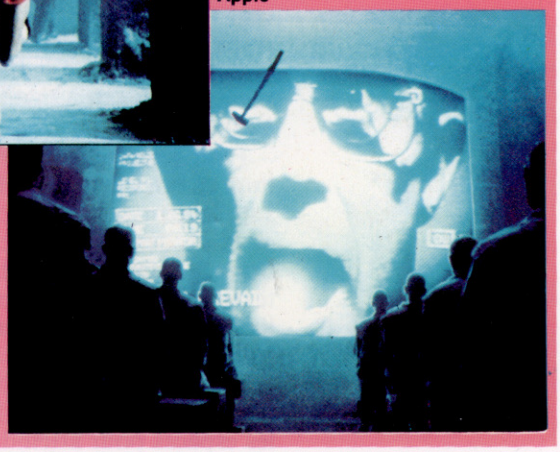
La conception et la diffusion des images des produits sont la partie créative du marketing (c'est l'avis des professionnels de la publicité). Mais la logistique est tout aussi importante — peut-être même plus, puisque c'est souvent à ce niveau que des problèmes se posent. En effet, c'est une chose de susciter l'intérêt pour un nouveau produit; mais acheminer ce produit dans les foyers et les bureaux est une tâche encore plus difficile.

La machine doit-elle être distribuée en vente par correspondance? C'est un bon moyen économique de livrer les ordinateurs aux clients. Mais comment assurer un bon service après-vente? Il est alors nécessaire de vendre au rabais pour contrer l'assurance que donne au client un achat en boutique; mais si vous réduisez trop vos marges, vous aurez du mal à financer une production massive, créant ainsi des retards dans les livraisons.

Dans la lutte pour la survie dans l'industrie de la micro-informatique, l'échec de certains fabricants d'excellents produits (logiciels et matériels) a démontré que la fabrication ne constitue que la moitié de la bataille. Créer et maintenir un marché est la tâche la plus importante et la plus risquée.

Acorn/BBC

Apple



Qualité courrier

Les imprimantes à marguerite produisent une impression d'une qualité bien supérieure à leurs rivales matricielles et offrent des fonctions très utiles. Leurs handicaps : coût et lenteur.

A première vue, il peut sembler étrange qu'un propriétaire d'ordinateur domestique désire acheter une imprimante à marguerite. Elles ne sont pas réellement destinées à l'impression de listages; elles sont lentes et plus chères que les imprimantes matricielles. Néanmoins, pour certaines applications, elles représentent un bon choix. Elles démontrent surtout leur supériorité lorsque la qualité d'impression est importante. Une imprimante matricielle forme chaque lettre en imprimant une configuration de points; les points individuels sont visibles dans le texte imprimé.

En revanche, avec une imprimante à marguerite, les caractères sont produits par un bloc de frappe qui heurte le ruban encré — tout comme une machine à écrire. Ces blocs de frappe, un pour chaque caractère, sont disposés en cercle, comme les pétales d'une fleur. D'où le nom « imprimante à marguerite ». L'impression qui en résulte est plus facile à lire et présente un aspect plus « professionnel ».

Mais cette meilleure qualité d'impression se paye par une vitesse moindre : les imprimantes à marguerite sont beaucoup plus lentes que les matricielles d'un prix comparable. Pour imprimer un caractère avec l'imprimante à marguerite, il est d'abord nécessaire de placer le bon pétale. Le marteau heurte ensuite le bloc d'impression et le chariot se déplace pour produire le caractère suivant.

Si l'on compare ce processus avec celui d'une imprimante matricielle où les points sont imprimés pendant le déplacement du chariot contre le papier, la différence de vitesse existant se comprend aisément. Une imprimante à marguerite de 5 000 F imprime environ 20 caractères à la second (cps); une imprimante matricielle coûtant environ le même prix a une vitesse d'impression de 160 cps. Certaines imprimantes à marguerite plus rapides sont aussi plus coûteuses — plus de 12 000 F pour une imprimante à 80 cps.

Pour augmenter la vitesse d'impression, ces imprimantes ont souvent deux caractéristiques supplémentaires : l'impression bidirectionnelle et la recherche logique. L'impression bidirectionnelle signifie simplement qu'une ligne de texte est imprimée de gauche à droite et que la ligne suivante est imprimée de droite à gauche. L'imprimante n'a pas à attendre le retour du chariot, elle imprime donc plus rapidement. La recherche logique signifie que le chariot passe les espaces pour atteindre le prochain mot dans le texte. Les imprimantes moins évoluées prennent le même temps pour imprimer un espace que pour tout autre caractère.

Les configurations des caractères des imprimantes matricielles sont stockées en ROM à l'intérieur de l'imprimante; les formes des caractères des imprimantes à marguerite sont simplement gravées sur les pétales de la marguerite. Chaque méthode a son avantage : avec une imprimante matricielle, les formes des caractères peuvent être redéfinies en envoyant les codes adéquats à l'imprimante à partir de votre micro. Avec une imprimante à marguerite le processus est beaucoup plus simple : il suffit de changer de roue d'impression.

Il existe une grande variété de roues d'impression offrant divers types de caractères et divers espacements de caractères. Parmi les types de caractères les plus utilisés, mentionnons le Courrier, le Roman, le Gothique et l'Italique. Une marguerite en plastique coûte environ 60 F, tandis que la métallique coûte plus de 240 F. Question de longévité !

Ces différents types de caractères posent un problème : peu d'entre eux utilisent exactement le même jeu de caractères que le micro. Cela signifie que certains caractères du clavier de votre micro seront imprimés différemment. Souvent, le caractère dièse (♯) est imprimé comme un symbole de livre anglaise (£) ou bien un crochet est imprimé comme une fraction (1/2). Avec de nombreux styles de caractères, le chiffre « 0 » (zéro) et la lettre majuscule « O » ne peuvent être distingués et, similairement, la lettre minuscule « l » peut être confondue avec le chiffre « 1 ». Les imprimantes à marguerite les plus coûteuses ont des roues d'impression de 127 caractères mais la plupart ne peuvent imprimer que 92 ou 97 caractères.

Comme vous pouvez l'imaginer, il peut être assez compliqué de corriger un listage comportant des caractères prêtant ainsi à confusion. Pour cette raison, et aussi à cause de sa lenteur d'impression, une imprimante à marguerite n'est pas recommandée si vous n'utilisez votre ordinateur que pour la programmation.

Une imprimante à marguerite peut être programmée pour produire une variété d'effets spéciaux, comme une imprimante matricielle. Bien que le nombre de ces effets soit limité, la méthode de programmation est identique. Elle consiste à envoyer des codes « escape ». Par exemple, sur une imprimante à marguerite Diablo, le code ESC-E sollicite le soulignement

automatique, et ESC-R y met fin. A l'aide du BASIC Microsoft, vous devez taper LPRINT CHR$(27);«E»; et LPRINT CHR$(27);«R»; pour envoyer ces codes à l'imprimante. ESC-I pose une tabulation; ESC-9 pose la marge gauche et ESC-U fait avancer le papier d'une demi-ligne vers le haut, très pratique pour imprimer des indices. Pour mettre certains caractères en valeur avec une imprimante à marguerite, on imprime chaque caractère à quatre reprises. Le code Diablo pour cette fonction est ESC-O. L'utilisation de tels caractères gras ralentit considérablement la vitesse d'impression.

Certaines imprimantes à marguerite vous permettent de faire varier à la fois les espaces entre les caractères et les interlignes. Si c'est le cas, l'imprimante peut être utilisée pour créer des images graphiques ou des copies d'écran. Si un point d'écran est illuminé, l'imprimante à marguerite imprime un point, si le point est éteint, un espace est imprimé. En réduisant la distance entre les caractères, une ligne complète d'écran peut être imprimée sur papier. De même, on peut diminuer l'espacement entre les lignes pour resserrer le texte.

Une des fonctions que nous n'avions pas vue sur les imprimantes matricielles est la possibilité de centrer les titres automatiquement. L'envoi d'un code ESC = à une imprimante Diablo centrera automatiquement le reste de cette ligne entre les marges. La tabulation décimale est une autre nouvelle fonction : après l'envoi d'un code ESC-H, tous les points décimaux des nombres décimaux sont alignés à l'impression, ce qui est très pratique pour des additions de sommes d'argent.

Les imprimantes à marguerite les plus coûteuses peuvent généralement offrir un espacement proportionnel des caractères. Avec cette fonction, l'espacement n'est pas de 4 ou 5 caractères au centimètre. Il est fonction de la largeur du caractère imprimé. Par exemple, le caractère « w » est beaucoup plus large que le caractère « i », le caractère « i » occupera donc moins d'espace que le caractère « w ».

L'acquisition, par une entreprise, d'une imprimante à marguerite peut être facilement justifiée; ce n'est probablement pas le cas pour la plupart des propriétaires d'ordinateurs domestiques. Reste la possibilité de connecter une machine à écrire électronique. Les imprimantes à marguerite sont plus coûteuses que les machines à écrire électroniques, même si elles utilisent la même méthode d'impression. Il y a encore peu de temps, les machines à écrire ne pouvaient pas facilement être connectées à un ordinateur — elles ne possédaient pas d'interface ou de prise RS232. Mais aujourd'hui, toutes les machines à écrire électroniques sont livrées avec une interface intégrée, ou peuvent en être équipées. Adapter une machine à écrire électronique permet d'économiser ainsi environ 2 400 F sur le prix d'une imprimante à marguerite et offre l'avantage de pouvoir toujours s'en servir comme d'une machine à écrire conventionnelle.

Le prix de la qualité

Exemple de frappe à grand espacement
Exemple de frappe à espacement moyen
Exemple de frappe à petit espacement

Un avancement d'un demi-interligne sert à imprimer des indices :

H_2O

ESC-E met en fonction le soulignement automatique et ESC-R le met hors fonction.

L'impression en caractères gras **fait ressortir** certains caractères du texte.

Le code ESC = sert à centrer du texte :

Un texte centré

Ces lignes de texte sont imprimées sans faire appel à l'espacement proportionnel. Notez comment les chiffres 0123456789 sont espacés. Sans espacement proportionnel, la largeur des caractères est toujours la même. Par exemple, un "W" occupe le même espace qu'un "i".

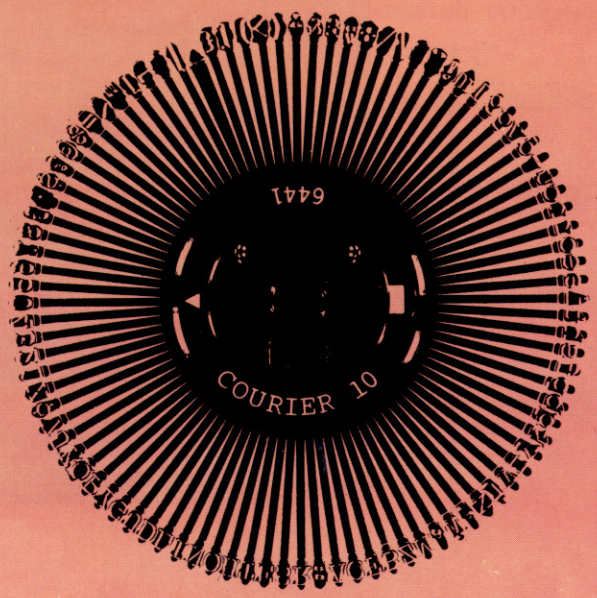
WWWWWWWWWWW
i i i i i i i i i i i

Ces lignes de texte sont imprimées en utilisant l'espacement proportionnel. Notez comment sont espacés les chiffres 0123456789. Avec l'espacement proportionnel, la largeur de chaque caractère est variable. Par exemple, un "W" est beaucoup plus large qu'un "i".

W W W W W W W W W W W
iiiiiiiiiii

L'effet global est de rendre le texte beaucoup plus agréable à lire.

N.B. – Pour des raisons techniques, nous avons dû remplacer les caractères machine par des caractères typographiques approchants.



Plus coûteuse et moins souple qu'une imprimante matricielle, l'imprimante à marguerite produit une impression d'une qualité comparable à celle d'une machine à écrire et un espacement proportionnel. La roue d'impression est facilement interchangeable.

Tout sur le clavier

Lorsque Sinclair Research a développé le basic du Spectrum, il a été prévu d'emblée de laisser la possibilité aux programmeurs expérimentés de contourner certaines de ses limites.

La touche !
Le clavier du Spectrum comprend 8 blocs de 5 touches, chacun étant constamment sous le contrôle d'un octet ou port qui lui est spécifique. Chaque touche est représentée par un bit du port. Lorsque l'on appuie sur une touche, son bit de signalement passe de 1 (son état normal) à 0. Ici les touches Y, I et K sont appuyées, ce qui donne une valeur binaire pour leur port, l'octet 57342, de XXX01001, les bits 5 et 7 étant indéfinis. De manière similaire, la frappe de A et de S donnera la valeur XXX11100 à l'octet 65022. (Cl. Kevin Jones.)

Le BASIC vous permet d'entrer des informations à partir du clavier par le biais de INPUT et de INKEY$. L'instruction INPUT permet de lire un nombre ou une chaîne de caractères terminés par ENTER, une variable numérique ou de chaîne. La fonction INKEY$ balaye le clavier et renvoie soit la chaîne qui comporte le caractère ou la touche demandée, soit, s'il n'est pas demandé de caractères, une chaîne vide. INPUT et INKEY$ sont deux commandes très utiles, mais lorsque des combinaisons de touches doivent être lues simultanément par exemple, la fonction IN permet de lire directement le clavier.

Les touches du clavier du Spectrum sont connectées au microprocesseur Z80 par l'intermédiaire de ports d'entrée/sortie. Il en existe 65536, et chacun peut être adressé séparément. De la même manière que PEEK et POKE servent à lire et à écrire en mémoire, IN et OUT servent à lire et à écrire aux ports d'entrée/sortie. La fonction IN(m) retourne la valeur du port *m,* et OUTPUT(m,n) écrira la valeur *n* sur le port *m*.

Le clavier consiste en quatre rangées de 10 touches, chacune étant divisée en deux demi-rangées de 5 touches. Chaque demi-rangée est représentée sur un port.

Les cinq bits les plus à droite d'un port codent une touche. Ils prennent la valeur 0 si la touche en question est frappée, et la valeur 1 lorsqu'elle ne l'est pas. Les bits marqués X ne sont pas spécifiés; ils peuvent contenir soit un un, soit un zéro. Cela signifie que la valeur IN(m) sera alors indéfinie. Ce problème peut être surmonté en affectant dans le logiciel la valeur 0 aux trois bits supérieurs, par l'instruction :

```
DEF FN a(p) = p - INT(p/32)*32
```

où p est la valeur du port (l'octet indéfini donné par IN(m)), et p/32 donnera un nombre compris entre 0 et une valeur inférieure à 8. INT(p/32) éliminera la partie fractionnaire de ce nombre, laissant une valeur entière comprise entre 0 et 7. En multipliant ce résultat par 32, on obtient la valeur qui représente les trois bits supérieurs ou les plus à gauche dans p. En soustrayant cette

valeur à p, on supprime ces trois bits, laissant une valeur entière comprise entre 0 et 31, qui représente les cinq bits inférieurs (les plus à droite) de p.

Par exemple, lorsque l'on frappe la touche V, le port 65278 prendra la configuration suivante :

X	X	X	0	1	1	1	1

FN a(IN(65278)) donnera donc la valeur :

0 0 0 0 1 1 1 1= 15

Si l'on frappe simultanément CONTROL et V, le port deviendra :

X	X	X	0	1	1	1	0

et FN a(IN(65278)) donnera la valeur 14. En l'absence de frappe, le port prendra la valeur 31 (00011111).

Essayez le programme suivant :

```
10 REM Afficher les valeurs masquées des ports
20 REM Définir la fonction de masque
30 DEF FN a(p)=p - INT(p/32)*32
40 REM Afficher les ports
50 PRINT AT 1,1; « Valeur masquée du port »
60 PRINT "32766", FN a(IN(32766))
70 PRINT "49150", FN a(IN(49150))
80 PRINT "57342", FN a(IN(57342))
90 PRINT "61438", FN a(IN(61438))
100 PRINT "63486", FN a(IN(63486))
110 PRINT "64510", FN a(IN(64510))
120 PRINT "65022", FN a(IN(65022))
130 PRINT "65278", FN a(IN(65278))
140 FOR i=1 TO 250: NEXT i
150 CLS
160 GO TO 50
```

Après avoir tapé RUN, faites des essais avec diverses frappes de touches et voyez comment les valeurs des ports changent. Essayez ensuite de prédire les valeurs affichées pour une touche, plusieurs touches et des combinaisons de touches. Si vous frappez les touches de déplacement du curseur (5, 6, 7 et 8), vous obtiendrez respectivement les valeurs masquées suivantes sur les ports 61438 et 63486 (que nous appellerons ports A et B).

	Port A (port 61438)	Port B (port 63486)
←	31	15
↓	15	15
↑	23	15
→	27	15

La touche CONTROL vous donnera la valeur masquée 30 pour le port 65278 (que nous appellerons port C).

Étudions le simple programme graphique donné ici. Il lit les touches de déplacement du curseur afin de tirer des lignes horizontales, verticales et diagonales et, en conjonction avec la touche CONTROL, déplace un « crayon graphique » sans tirer de ligne. Les lignes diagonales s'obtiennent en frappant deux touches de déplacement de curseur à la fois.

La ligne 30 du programme définit une fonction FN(a) qui masque les trois bits supérieurs, de la même manière que précédemment. La position horizontale du crayon optique est représentée par *x*, *y* étant sa coordonnée verticale. La ligne 50 initialise la position du crayon sur le centre de l'écran.

Les lignes 70 et 80 lisent les ports attribués aux mouvements du curseur. La demi-rangée comprise entre 6 et 0 comprend trois des touches du curseur; elle est représentée sur le port 61438 (port A). La demi-rangée comprise entre 1 et 5 se rapporte à la touche de curseur « déplacement vers la droite »; elle correspond au port 63486 (port B). La ligne 90 lit la touche CONTROL, la demi-rangée de V à CONTROL étant représentée sur le port 65278 (port C).

Les lignes 110 à 180 testent les 8 déplacements possibles du curseur, la position (*x*, *y*) étant adaptée en conséquence.

La ligne 110 teste la flèche ↑
La ligne 120 teste les flèches ↑ et →
La ligne 130 teste →
La ligne 140 ↓ et →
La ligne 150 ↓
La ligne 160 ↓ et ←
La ligne 170 ←
La ligne 180 ↑ et ←

Les lignes 200 à 240 font en sorte que le « crayon » ne sorte pas de l'écran.

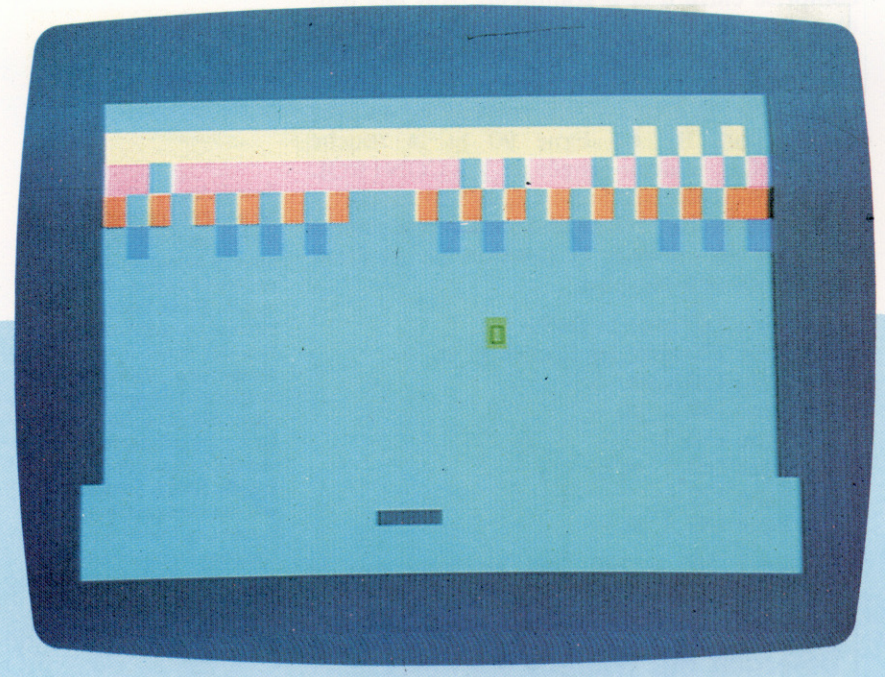
Double jeu
Le programme met en œuvre les procédés décrits pour la détection des doubles frappes : la frappe simple d'une touche curseur vous permet de tirer des lignes horizontales, verticales ou diagonales (diagonales = horizontales + verticales). La frappe conjuguée d'une touche curseur et de la touche CONTROL produit les mêmes déplacements, mais sans tracés à l'écran. Dans un prochain article nous appliquerons ce principe plus efficacement.

Détection des frappes multiples

```
20  REM Fonction masque des trois bits supérieurs
30  DEF FN a(p)=p-INT (p/32)*32
40  REM Initialisation de la position du crayon
50  LET x=127: LET y=35
60  REM Lire les ports et masquer les 3 bits supérieurs.
70  LET por tA=FN a(IN 61438)
80  LET por tB=FN a(IN 63436)
90  LET por tC=FN a(IN 65278)
100 REM Changer la position du crayon selon la frappe
110 IF por tA=23 AND portB=31 THEN LET y=y+1
120 IF por tA=19 AND portB=31 THEN LET x=x+1:
    LET y=y+1
130 IF por tA=27 AND portB=31 THEN LET x=x+1
140 IF por tA=11 AND portB=31 THEN LET x=x+1:
    LET y=y-1
150 IF por tA=15 AND portB=31 THEN LET y=y-1
160 IF por tA=15 AND portB=15 THEN LET x=x-1:
    LET y=y-1
170 IF por tA=31 AND portB=15 THEN LET x=x-1
180 IF por tA=23 AND portB=15 THEN LET y=y+1:
    LET x=x-1
190 REM Empêcher le crayon de sortir de l'écran
200 IF x 0 THEN LET x=0
210 IF x 255 THEN LET x=255
220 IF y 0 THEN LET y=0
230 IF y 175 THEN LET y=175
240 REM Trace le point si CONTROL n'a pas été frappé.
250 IF por tC=31 THEN PLOT x, y
260 GO TO 70
```

Casse-briques

Voici un jeu que l'on a l'habitude de voir dans les cafés... et dans les fêtes foraines. Sans dépenser un centime, Pierre Monsaut vous propose ce programme écrit pour le micro Dragon.



Le but du jeu est simple : essayer de détruire un mur de brique à l'aide d'une balle que vous devez renvoyer avec votre raquette. Chaque brique cassée rapporte un point. Lorsque le mur est entièrement détruit, un nouveau mur apparaît. Vous disposez de dix balles pour tenter de marquer un score maximal. Utilisez le joystick ou les touches A, S et la barre d'espacement pour déplacer la raquette.

```
10 REM *****************
20 REM * CASSE-BRIQUES *
30 REM *****************
40 S=0:NB=0
50 GOSUB 530
60 V=V+DV:H=H+DH
70 POKE B+1024,CN
80 B=V*32+H
90 L=PEEK(B+1024)
100 IF L<>223 AND L<>128 THEN DV
=-DV:K=0:S=S+1
110 POKE B+1024,CB
120 IF V=13 AND ABS(R-29-B)>1 TH
EN 320
130 IF V=13 AND H>2 AND H<29 THE
N POKE B+1024,CN:H=H+CH
140 IF V=13 OR V=1 THEN DV=-DV
150 IF H=1 OR H=30 THEN DH=-DH
160 ON JS GOSUB 220,250
170 IF R<446 THEN R=446
180 IF R>475 THEN R=475
190 PRINT@ R,R$;
200 IF S/120=INT(S/120) AND K=0
THEN GOSUB 670
210 GOTO 60
220 L=INT(JOYSTK(0)/2+445)
230 IF ABS(L-R)>1 THEN R=R+2*SGN
(L-R):CH=SGN(L-R)
240 RETURN
250 D$=INKEY$
260 D=2*((D$="A")-(D$="S"))
270 IF D$=" " THEN DO=0
280 IF D<>0 THEN DO=D
290 R=R+DO
300 CH=SGN(DO)
310 RETURN
320 NB=NB+1
330 FOR I=1 TO 5
340 SOUND 1,1
350 FOR J=1 TO 50
360 NEXT J,I
370 IF NB=11 THEN 410
380 POKE B+1024,CN
390 GOSUB 760
400 GOTO 60
410 IF S>R1 THEN R1=S
420 PRINT@ 166,"SCORE :";S;
430 PRINT@ 230,"RECORD :";R1;
440 PRINT@ 294,"UNE AUTRE ?";
450 FOR I=1 TO 100
460 D$=INKEY$
470 NEXT I
480 D$=INKEY$
490 IF D$="" THEN 480
500 IF D$<>"N" THEN 40
510 CLS
520 END
530 CLS
540 PRINT@ 203,"JOYSTICK ?";
550 D$=INKEY$
560 IF D$="" THEN 550
570 IF D$="O" THEN JS=1 ELSE JS=
2
580 CLS 6
590 R=461
600 FOR I=1024 TO 1055
610 POKE I,128
620 NEXT I
630 FOR I=1 TO 12
640 POKE I*32+1024,128
650 POKE I*32+1055,128
660 NEXT I
670 FOR I=1057 TO 1086
680 POKE I+32,159
690 POKE I+64,239
700 POKE I+96,255
710 POKE I+128,175
720 NEXT I
730 K=1
740 R$=CHR$(223)+CHR$(223)+CHR$(
211)+CHR$(211)+CHR$(211)+CHR$(22
3)+CHR$(223)
750 CB=79:CN=223
760 V=13:DV=-1
770 H=RND(28)+1
780 B=V*32+H
790 DH=(RND(2)-1.5)*2
800 B1=B:DO=0
810 RETURN
```

Extension mémoire

Le BBC Micro est très limité par sa capacité mémoire. Pour les usages professionnels, le Torch Disk Pack pallie précisément à cet inconvénient en apportant des extensions.

Le BBC Micro peut, par extension, atteindre le standard professionnel. Bien que cette possibilité ait été prévue dès le départ, sa réalisation n'est pas le fait du constructeur du BBC, Acorn, mais d'une société étrangère à ce micro, Torch. Le Torch Disk Pack permet à l'utilisateur du BBC d'exploiter la vaste gamme de logiciels qui utilisent le système d'exploitation CP/M. Le prix est un peu plus élevé que celui d'un double lecteur de disques Acorn.

Le système d'exploitation CP/M nécessite un microprocesseur Z80 et un double lecteur de disques. Le BBC utilise un microprocesseur 6502, et l'adjonction du Z80 relègue le 6502 à la seule gestion des fonctions d'entrée/sortie. Le clavier, l'écran et les lecteurs sont sous le contrôle du 6502; le Z80 avec ses 64 K de RAM exploite les programmes sous CP/M et prend en charge le système. Les données circulent d'un processeur à l'autre par le biais du port d'extension du BBC appelé aussi « tube ».

Le processeur Z80 et l'extension mémoire qui l'accompagne figurent sur une petite carte imprimée qui prend place dans le BBC. L'alimentation du BBC n'est pas sollicitée par le système ainsi assemblé. Une nouvelle alimentation qui se trouve dans le boîtier du chargeur de disques est connectée à l'ordinateur. Le boîtier comporte également le double-lecteur. Il s'agit de deux lecteurs 80 pistes double-face très semblables aux lecteurs de 800 K Acorn pour le BBC.

Torch présente en alternative un autre système d'extension pour le BBC : le ZEP 100, identique au Torch Disk Pack, à la différence près qu'il ne comprend pas de lecteurs. Il est donc destiné aux utilisateurs de BBC qui disposent déjà des lecteurs de disques. Torch commercialise également une gamme d'ordinateurs professionnels composés d'une carte BBC et de lecteurs de disques, ainsi que d'un moniteur et d'un modem.

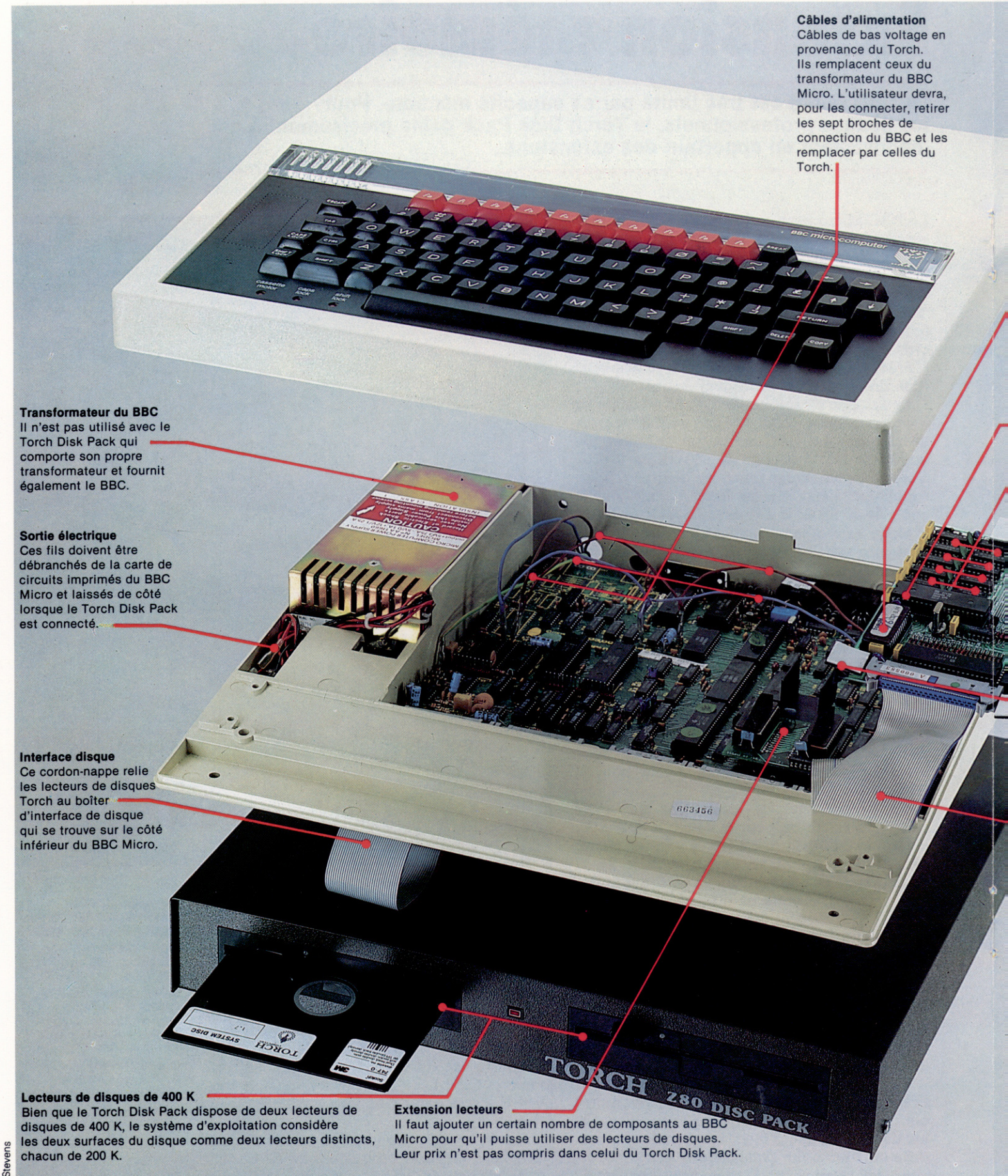
Le chargeur de disques est conçu pour que le BBC soit posé dessus, de courts câbles assurant la connection. Un ennui : le chargeur soulève le clavier de manière intempestive pour ce qui est de la frappe.

Le Torch Disk Pack a tout prévu pour assurer l'extension du BBC, à une seule exception près qui est de taille, pourtant : il ne fournit pas le jeu de composants d'interfaçage disques nécessaire au BBC pour exploiter des lecteurs. Certains utilisateurs les auront peut-être acquis avec leur micro, mais la plupart auront acheté un BBC standard Model B, et devront se les procurer. Ce coût doit donc être ajouté à celui du chargeur.

Extension du BBC
Le Torch Disk Pack fait du Micro BBC un ordinateur professionnel par le processeur Z80 qu'il lui apporte et l'extension mémoire de 64 K qui lui permet d'utiliser un système d'exploitation proche du CP/M. Il comprend en outre deux lecteurs de 400 K, le tout pour un prix à peine supérieur à celui du double-lecteur Acorn.
(Cl. Chris Stevens.)

Une fois le système d'extension connecté et interfacé, il se met en marche directement à l'allumage (en mode d'affichage 3) tout comme un ordinateur sous CP/M. Le système d'exploitation du disque a été écrit par Torch et s'appelle MCP. Il est très semblable au CP/M et pourra exploiter des centaines de programmes CP/M. La différence principale étant que MCP réside en ROM dans le BBC et n'a pas à être chargé.

De nombreuses commandes du MCP paraîtront familières aux utilisateurs du BBC. MODE permet de choisir différents modes d'affichage, et les commandes relatives à l'unité de visualisa-



Câbles d'alimentation
Câbles de bas voltage en provenance du Torch. Ils remplacent ceux du transformateur du BBC Micro. L'utilisateur devra, pour les connecter, retirer les sept broches de connection du BBC et les remplacer par celles du Torch.

Transformateur du BBC
Il n'est pas utilisé avec le Torch Disk Pack qui comporte son propre transformateur et fournit également le BBC.

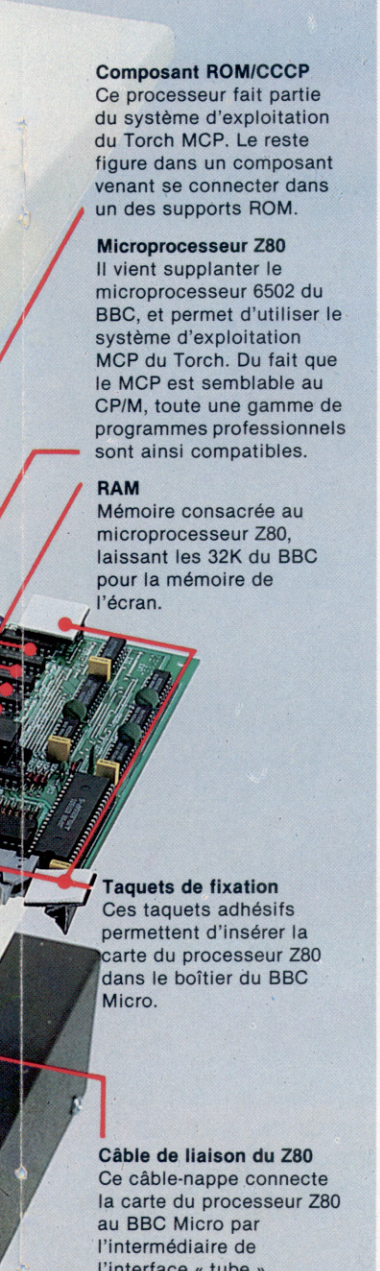
Sortie électrique
Ces fils doivent être débranchés de la carte de circuits imprimés du BBC Micro et laissés de côté lorsque le Torch Disk Pack est connecté.

Interface disque
Ce cordon-nappe relie les lecteurs de disques Torch au boîter d'interface de disque qui se trouve sur le côté inférieur du BBC Micro.

Lecteurs de disques de 400 K
Bien que le Torch Disk Pack dispose de deux lecteurs de disques de 400 K, le système d'exploitation considère les deux surfaces du disque comme deux lecteurs distincts, chacun de 200 K.

Extension lecteurs
Il faut ajouter un certain nombre de composants au BBC Micro pour qu'il puisse utiliser des lecteurs de disques. Leur prix n'est pas compris dans celui du Torch Disk Pack.

Chris Stevens



Composant ROM/CCCP
Ce processeur fait partie du système d'exploitation du Torch MCP. Le reste figure dans un composant venant se connecter dans un des supports ROM.

Microprocesseur Z80
Il vient supplanter le microprocesseur 6502 du BBC, et permet d'utiliser le système d'exploitation MCP du Torch. Du fait que le MCP est semblable au CP/M, toute une gamme de programmes professionnels sont ainsi compatibles.

RAM
Mémoire consacrée au microprocesseur Z80, laissant les 32K du BBC pour la mémoire de l'écran.

Taquets de fixation
Ces taquets adhésifs permettent d'insérer la carte du processeur Z80 dans le boîtier du BBC Micro.

Câble de liaison du Z80
Ce câble-nappe connecte la carte du processeur Z80 au BBC Micro par l'intermédiaire de l'interface « tube ».

tion ainsi que celles de type FX sont toutes disponibles. KEY sert ici à définir également les touches de fonction. Quatre touches de fonction sont prédéfinies par des commandes.

Il existe de nombreuses autres commandes pour le chargement et la gestion des fichiers sur disque. Outre qu'il permet de charger des programmes en code machine simplement en tapant leur nom, le système propose une autre commande pour charger des fichiers spéciaux qui élaborent sur disque des figures graphiques en utilisant le système d'exploitation du BBC. PRINT permet d'imprimer un fichier texte, et TYPE de l'afficher. COMMAND utilise un fichier comme une série de commandes pour l'ordinateur, de manière similaire à EXEC pour le BASIC du BBC.

Il existe d'autres programmes utilitaires qui figurent sur un « disque système » livré avec le chargeur. Ils se chargent en tapant simplement leur nom. Parmi eux, une routine pour modifier la présentation des caractères à l'écran, un utilitaire pour écrire de la musique, un dépisteur d'erreur de code machine, un éditeur de disque, et un utilitaire permettant de lire les disques Acorn et réciproquement. Ce dernier est très utile dans la mesure où les formats des deux systèmes sont différents. Il vous permet par exemple de produire un fichier texte, à partir d'un programme BASIC BBC, pour saisir du texte et l'éditer au moyen d'un programme de traitement de texte sous CP/M.

Bien que le format-disque du Torch soit différent de celui d'Acorn, il utilise la même convention bizarre qui consiste à considérer les deux lecteurs de disques comme quatre lecteurs distincts. Le BBC Micro, ainsi transformé en configuration « étendue », peut toujours être utilisé comme un micro standard. En tapant BASIC, le système ignorera le chargeur et ses ressources. Curieusement, le BBC Micro qui en est dépourvu est conçu pour utiliser des cassettes et non des disquettes. Malgré le fait que le micro est interfacé pour accéder aux lecteurs du chargeur, en l'absence de MCP, l'utilisateur devra spécifier que le système de gestion de fichier sur disque est appelé. Le BASIC BBC vous permet de revenir sous Z80 à tout moment, pourvu que vous éteigniez la machine, la rallumiez, ou tout simplement en frappant *MCP.

Le micro-ordinateur BBC est doté de nombreuses facilités, mais il reste un ordinateur domestique. Pour les applications professionnelles, il se révèle insuffisant. Avec l'extension Torch Disk Pack, il acquiert les caractéristiques essentielles d'une machine professionnelle.

Adaptation du chargeur de disques
L'adaptation du chargeur au BBC Micro n'est pas simple. Votre ordinateur doit être un BBC Micro Modèle B doté d'une interface lecteur de disques. Comme l'alimentation interne du BBC n'est plus nécessaire, il vous faut retirer les sept broches d'alimentation sur la carte principale sur laquelle vous brancherez le câble en provenance du Torch.
Un câble-nappe vient s'insérer dans l'interface lecteur de disques du BBC. Le composant de ROM où figure le système d'exploitation MCP est alors connecté dans un support disponible de la ROM du BBC.

Chris Stevens

Logiciels gratuits
Le Torch est livré avec plusieurs logiciels. Un ensemble de progiciels de gestion : Perfect Writer (programme de traitement de texte), Perfect Speller (vérification de l'orthographe), Perfect Filer (programme de bases de données) et Perfect Calc (programme de tableaux électroniques).
Outre ces progiciels, une version du BBC BASIC est fournie pour le processeur Z80. Toutes les commandes sont identiques à celles figurant en ROM dans le BBC. La seule exception étant l'assembleur intégré 6502 du BBC. En effet, le processeur du Torch étant du type Z80 et non 6502, le BBC BASIC Z80 comprend un assembleur Z80. Le gain de mémoire, quel que soit le mode d'affichage choisi, sera de 48 K. Le BBC Micro par lui-même ne dispose jamais que de 9 K de libres pour certains modes d'affichage, et n'a jamais plus de 28 K de libres.

Composants DFS
Un certain nombre de composants doivent être ajoutés au BBC pour lui permettre d'utiliser des lecteurs de disques. Ils doivent être acquis séparément du Torch Disk Pack et leur coût vient donc s'ajouter. Certaines de ces extensions ne concernent pas spécifiquement le Torch. Ainsi le composant DFS, dont le chargeur ne se sert pas. Il est cependant utile (remplissage du disque) dans la mesure où le DFS est le système d'exploitation utilisé par Acorn. Ce qui signifie que le Torch peut être utilisé en tant que double lecteur.
Les formats disques Torch et Acorn ne sont pas compatibles, aussi les programmes écrits sous le système d'exploitation MCP ne peuvent lire des disques Acorn, et réciproquement. Le Torch est livré avec un programme qui assure le passage entre les deux formats.

Options professionnelles
Le premier ordinateur Torch était une machine professionnelle de coût élevé. Torch produit également des versions professionnelles utilisant le système d'exploitation UNIX. Il s'agit de la série 700 (sur la photo). Il existe aussi des terminaux (série 300) qui réunissent plusieurs micros Torch en un réseau.

Couleurs et nombres

Nous vous proposons aujourd'hui un programme s'inspirant de ce jeu bien connu de cours de récréation et de camps de vacances, « Jacques a dit... ».

« Jacques a dit... » est l'un des premiers jeux auxquels on ait joué. Le meneur de jeu donne des instructions telles que « Jacques a dit : mettez vos mains sur la tête! » ou « Jacques a dit : levez-vous! ». Les joueurs sont disqualifiés ils ne réagissent pas en conséquence.

Ce jeu pourrait être mis en œuvre dans un jeu électronique utilisant un microprocesseur pour contrôler des commandes de jeu, des lumières et des sons appropriés. L'ordinateur serait alors le meneur de jeu dont les instructions devraient être suivies à la lettre.

Le jeu électronique émettrait des séquences sonores et de couleurs que le(s) joueur(s) devrai(en)t répéter en appuyant sur les touches appropriées.

Vous pouvez mettre à profit votre ordinateur pour réaliser vous-même ce jeu, et à votre manière. En effet, même un jeu attrayant finit par lasser, ce qui n'est pas le cas avec l'ordinateur lorsqu'il est programmé de manière riche et subtile. En outre, un micro n'est pas limité comme un jeu électronique, dont la capacité de traitement reste faible.

Le programme que nous donnons ici s'appelle « Répétez! ». Il est fondé sur quatre couleurs, chacune dotée d'une sonorité propre et d'une touche spécifique au clavier (numérotées de 1 à 4). Le programme affiche une couleur et vous demande de la reproduire au clavier. Si vous y parvenez, le programme affiche alors deux couleurs. Si à nouveau vous savez les restituer, il vous en proposera trois, et ainsi de suite...

Vous pourrez perdre de deux manières : en tardant à répondre quand le rythme devient trop rapide ou en faisant trois fautes pour un jeu. Pour rendre le jeu encore plus difficile, le temps d'affichage devient progressivement plus court. Vous ne devez pas, pour autant, restituer la séquence de couleurs à la même vitesse qu'elle a été donnée. Le jeu peut proposer un maximum de 50 couleurs en une séquence. Il y a peu de chances que vous y parveniez. La plupart des joueurs suivent des cadences de 15 affichages, pas plus!

Étudions la structure du programme : toutes les commandes couleurs et sons sont regroupées à la fin du programme sous forme de sous-programmes, de la manière suivante :

```
1000 Afficher la couleur a
1500 Lire au clavier la frappe 1, 2, 3 ou 4
2000 Émettre un signal pour la fin du jeu
2500 Émettre un signal de mauvaise réponse
6000 Afficher un message à la ligne 20 et attendre si besoin est
```

Le fait que ces sous-programmes soient distincts du corps principal du programme présente deux avantages : d'abord tous les problèmes de sons et de couleurs, spécifiques à la machine, sont traités séparément, ce qui permet de passer le programme sur une autre machine; ensuite, cela permet de réutiliser les sous-programmes pour d'autres jeux apparentés. Le même programme peut ainsi rendre possible des variations sur le même jeu, de la même manière qu'un jeu électronique. Vous pouvez également inventer des raffinements de votre cru et les ajouter au programme. Par exemple, le programme peut être adapté afin d'autoriser l'affichage des scores individuels pour tout nombre de joueurs.

En tant qu'utilisateurs de micro, soyez conscients du fait que l'ordinateur peut toujours faire mieux et plus efficacement qu'un jeu électronique.

Kevin Jones

Répétez !

```
  10  REM Jeu "Répétez"
  30  LET h=0: LET n=0: LET w=3
  40  DIM c(4): DIM p(4): DIM a(50)
  50  LET p(1)=5: LET p(2)=8: LET p(3)=12: LET
p(4)=15
  60  LET c(1)=1: LET c(2)=2: LET c(3)=4: LET
c(4)=6
  70  LET s$=" ": FOR i=1 TO 5: LET s$=s$+s$:
NEXT i
  80  LET b$=CHR$ (143) : REM Un bloc
 100  REM *** Page d'instructions
 110  CLS : PRINT TAB (10) : "Répétez!"
 120  PRINT : PRINT
 130  IF n)0 THEN PRINT : PRINT "Vous avez tenu ":n:" fois"
 140  IF n)h THEN PRINT : PRINT "...Nouveau record!!!"
: LET h=n
 150  IF h)0 THEN PRINT : PRINT "Le meilleur score est
jusqu'à présent":n:" affichages"
 155  PRINT:PRINT "Essayez de reproduire la
séquence de lumières et de sons de
l'ordinateur"
 160  PRINT "en appuyant sur les touche 1
à 4"
 170  PRINT : PRINT "Appuyer sur P (Play)
pour jouer, S pour interrompre"
 180  LET a$=INKEY$: IF a$="" THEN GO TO 180
 190  IF a$="s" OR a$="S" THEN CLS : STOP
 200  IF a$()"p" AND a$  "P" THEN GO TO 180
 205  REM *** Nouveau jeu...
 210  CLS : PRINT TAB (10): "Répétez!"
 220  FOR a=1 TO 4: GO SUB 1000: NEXT a
 230  LET n=0: LET m=0: RANDOMIZE
 240  REM *** Jeu suivant
 250  LET n=n+1
 270  LET a(n)=INT (RND*4)+1
 280  IF m=w THEN GO SUB 2000: LET m$="*"+STR $(w)
+"mauvaises réponses!: GO SUB 6000: GO TO 100
 285  LET m$="*Et voilà ...": GO SUB 6000
 290  FOR i=1 TO n
 300  LET a=a(i): GO SUB 1000
 310  FOR j=i TO 100/n: NEXT j
 320  NEXT i
 330  LET m$="Répétez ...": GO SUB 6000
 340  LET i=1
 350  GO SUB 1500
 360  IF t=0 THEN GO SUB 2000: LET m$="*Too s
low!": GO SUB 6000: GO TO 100
 370  IF a()a(i) THEN LET m=m+1: GO SUB 2500:
 GO TO 280
 380  LET i=i+1: IF)i =n THEN GO TO 350
 390  IF n 50 THEN LET m$="*Vous avez gagné avec 50 affichages
!": GO SUB 6000: GO TO 100
 400  LET m$="* Prêt à recommencer": GO SUB
 6000
 410  GO TO 250
1000  REM *** couleur "a"
1010  INK c(a)
1015  LET p=(a-1)*8+2
1020  FOR i=10 TO 14
1030  PRINT AT i,p:
1040  IF i=12 THEN PRINT b$:b$:a:b$:b$:
1050  IF i()12 THEN PRINT b$:b$:b$:b$:b$:
1060  NEXT i
1070  BEEP 2/(n+1),p(a)
1080  PRINT AT 10,p:"    "
1090  PRINT AT 11,p:" ":b$:b$:b$:" ";
1100  PRINT AT 12,p:" ":b$:a:b$:" ";
1110  PRINT AT 13,p:" ":b$:b$:b$:" ";
1120  PRINT AT 14,p:"    "
1130  INK 0: RETURN
1500  REM *** Lire la frappe au clavier
1510  LET t=250
1520  LET a$=INKEY$: IF a$="" THEN LET t=t-1:
 IF t)0 THEN GO TO 1520
1530  IF t=0 THEN RETURN
1540  IF a$()"1" AND a$()"2" AND a$()"3" AND a
$()"4" THEN GO TO 1520
1550  LET a=VAL (a$): GO SUB 1000
1560  RETURN
2000  REM *** bruit provocateur
2010  BEEP 3.0: RETURN
2500  REM Signal d'avertissement
2510  BEEP 1.0: RETURN
6000  REM *** afficher m$
6010  PRINT AT 20 1:s$:AT 20,1:
6030  IF m$(1)="*" THEN PRINT m$(2 TO): FOR
z=1 TO 200: NEXT z
6040  IF m$(1)()"*" THEN PRINT m$
6050  RETURN
```

Variantes de basic

Commodore 64

Remplacez CLS par PRINT CHR$(147)
Remplacez LET A$=INKEY$ par GET A$
Remplacez RANDOMIZE par XX=RND(-TI)
Remplacez (RND*4) par (RND(1)*4)
Remplacez M$(1) par LEFT$(M$,1)
Remplacez M$(2 TO) par MID$(M$,2)
Remplacez CHR$(143) par CHR$(166)
Remplacez PRINT AT L,C; par PRINT LEFT$(DN$,L+1) TAB(C); (par exemple la ligne 6010 devient
6010 PRINT LEFT$(DN$,21)TAB(1);S$;LEFT$(DN$,21)TAB(1);
Remplacez DIM C(4) par DIM C$(4)
Remplacez b$;b$;a;b$;b$; à la ligne BBZBB$; à la ligne 1040
Remplacez b$;a;b$; par BZB$ à la ligne 1100.
Ajoutez :

```
20 VL=54296:AD=54277:SR=AD+1:WF=AD-1:NO=17:
N1=NO:LF=AD-5:HF=LF+1
25 POKE AD,255:POKE SR,48:POKE VL,15
50C$(1)=CHR$(31):C$(2)=CHR$(28):C$(3)=CHR$(30):C$(4)=
CHR$(158)
60 P(1)=51:P(2)=34:P(3)=64:P(4)=38
90 DN$=CHR$(17):FOR K=1 TO 5:DN$=DN$+DN$:NEXT K:
DN$=CHR$(19)+DN$
1010 PRINT C$(A);
1015 P=(A-1)*9+3:Z$=RIGHT$(STR$(A),1)
1030 PRINT CHR$(144):RETURN
2010 SD=15:SP=4:N1=33:GOSUB 7000:RETURN
2510 SD=10:SP=10:N1=33:GOSUB 7000:RETURN
7000 REM*** BIP SONORE SD, SP
7010 POKE VL,15:POKE WF,N1
7020 POKE LF,SP:POKE HF,SP:
FOR DD=1 TO SD*50:NEXT DD
7030 POKE HF,0:POKE LF,0:N1=NO:RETURN
```

BBC Micro

Remplacez AT Y,X par TAB(X,Y). Par exemple, la ligne 6010 devient
6010 PRINT TAB(1,20);S$;TAB(1,20)
Remplacez INKEY$ par INKEY$(0)
Ajoutez :

```
20 MODE 2
25 COLOUR 135:CLS
60 C(1)=1:C(2)=2:C(3)=4:C(4)=5
80 B$=CHR$(35)
1010 COLOUR C(A)
1015 P=(A-1)*4
1070 SOUND 1,-10,P(A),40/(N+1):FOR DE=1
TO 2000/N:NEXT
1130 COLOUR 0:RETURN
2010 SOUND 1,-15,2,40
2510 SOUND 1,-15,40,20
```

Plan d'action

Nous allons étudier ici la conception générale d'un programme et examiner des questions qui devraient être posées avant d'en écrire un. C'est une étape importante de la programmation.

La conception d'un programme est perçue par les personnes impliquées (le programmeur et l'utilisateur) comme une recherche de solutions à des problèmes précis.

Malheureusement, on a toujours tendance à croire que les problèmes à résoudre sont d'une nature technique — comment formater l'écran, comment accélérer l'exécution d'une boucle, comment tout loger dans la RAM et ainsi de suite — tandis que les véritables problèmes apparaissent dès le début du projet, et sont généralement créés lors de la première rencontre entre l'utilisateur et « l'expert ». Les utilisateurs peuvent rarement présenter clairement leurs besoins et indiquer comment les satisfaire — ils espèrent que les experts sauront découvrir ces besoins et la solution la mieux adaptée — et les experts pensent souvent connaître les besoins et la solution avant que l'utilisateur n'ait commencé à parler. Cela entraîne une mauvaise communication initiale.

Dans les projets de programmation sur ordinateur domestique, le programmeur est généralement à la fois l'analyste système et le client. Les problèmes de communication devraient être considérablements réduits. Néanmoins, en tant qu'utilisateur-concepteur, vous devriez toujours faire l'effort de vous expliquer la nature de vos besoins, les solutions à envisager et les exigences requises aussi clairement que si vous parliez à une autre personne.

Prenons un utilisateur fictif et son problème : un modéliste possède un micro-ordinateur muni d'une unité à cassette. Il désire stocker des descriptions assez détaillées des matériaux utilisés dans la construction de chaque modèle qu'il a réalisé de façon à retrouver dans ses enregistrements tel type de colle, ou tel type de joint. L'analyste doit alors obtenir auprès de l'utilisateur les éléments d'information suivants :

• La fonction du programme. Cela pourrait être, au début, une vague description de ses intentions comme : « Je désire garder des données concernant mes modèles »; mais le concepteur doit essayer de faire parler l'utilisateur afin de mieux connaître ses véritables besoins comme : « Le programme devrait pouvoir stocker mes descriptions du modèle, de sa construction et des matériaux utilisés, comme je les ai tapées au clavier, et afficher ces renseignements lorsque je tape le nom du modèle, ou un aspect quelconque de sa construction ». Les besoins de l'utilisateur sont ainsi formulés plus clairement, et les diverses tâches de programmation impliquées sont maintenant identifiées (stockage, recherche, indexation, extraction, etc.).

• Comment le programme sera utilisé. Certains des détails physiques d'un usage particulier ne peuvent pas être clairement déduits au moyen de la description de la fonction. Par exemple, il se peut que l'utilisateur ne désire pas que les détails concernant ses modèles soient affichés sur écran parce qu'il travaille sans moniteur dans un atelier. Dans ce cas, les renseignements devront plutôt être imprimés.

• Quels seront les formats d'entrée et de sortie. Le programmeur professionnel utilisera souvent des tableaux pré-imprimés représentant l'écran pour dessiner ce que l'utilisateur verra pendant les phases d'entrée et de sortie. Ce n'est pas vraiment nécessaire dans le cas d'une utilisation domestique, bien que les graphiques haute résolution puissent constituer une exception. Les formats d'écran représentent un aspect très important des relations entre l'utilisateur et la machine, et méritent une attention particulière (position du clavier et du moniteur, hauteur de la table, niveaux d'éclairage, etc.).

• Comment doivent être organisés les fichiers. L'utilisateur peut estimer qu'il a vraiment besoin de pouvoir stocker au moins cent descriptions d'avions. D'autre part, il se peut qu'il ne désire construire qu'une douzaine de modèles. La dimension des fichiers de données du programme détermine quels devront être leurs formats et leurs méthodes d'accès. Une recherche séquentielle dans six descriptions de modèles prenant environ cinq minutes pourrait être acceptable, tandis que le temps d'attente pour une recherche dans cent descriptions ne le serait pas. Une solution pourrait être de mettre un fichier index des descriptions sur une cassette, et les descriptions elles-mêmes sur vingt autres cassettes classées par type d'avion.

La taille du programme lui-même peut aussi devenir un problème : si la section de saisie de texte demande un éditeur de texte complexe, si le programme est déjà chargé de menus et de messages, si les sections de gestion de fichiers emploient des routines compliquées de recherche et d'indexation, le programme peut devoir être fractionné en plusieurs programmes séparés afin de pouvoir être logé en RAM.

• Que doit-on prévoir comme procédures spéciales, comme calculs? Dans notre exemple, cette question ne se pose pas vraiment, mais c'est souvent le cas lorsque d'autres problèmes doivent être solutionnés. Il peut exister jusqu'à

vingt méthodes excellentes pour effectuer un processus, mais l'utilisateur peut insister sur le choix de l'une d'entre elles.

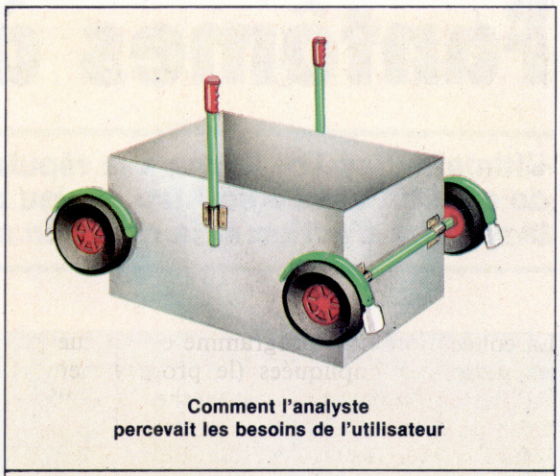
Ne pas respecter ce choix peut avoir pour conséquence que l'utilisateur ne sera jamais satisfait du programme. Le programmeur peut être tenté de choisir une approche différente de celle de l'utilisateur parce que plus efficace à son avis; mais les avantages ainsi gagnés sont rapidement perdus si l'utilisateur n'utilise pas le programme! Alors, il peut être très important de connaître les méthodes de travail de l'utilisateur. Pourquoi, par exemple, inventer une formule de calcul aéronautique, si vous pouvez simplement demander au constructeur de modèles comment il procède.

Après avoir noté toutes ces informations, la traduction de ces exigences dans un programme peut commencer. Une approche efficace consisterait à concevoir d'abord le dialogue programme-utilisateur, puis à penser aux fichiers de données et aux processus qui les gèrent. Le terme « dialogue » représente ici l'échange d'informations entre l'utilisateur et le programme. Cela n'implique pas uniquement l'entrée des détails concernant un avion et leur affichage subséquent, mais inclut aussi chaque message ou menu produit par le programme et chaque entrée, commande ou sélection faites par l'utilisateur. Il est aussi important d'établir à ce stade un style de dialogue. Pour ce programme, nous avons le choix entre un pilotage par commandes ou par menus. Les décisions prises ici auront un effet considérable sur la structure globale du programme. Le contenu et le format du dialogue doivent être examinés de façon détaillée, mais cet effort sera récompensé, puisque vous spécifiez ainsi toutes les données manipulées par le programme. L'espace de stockage requis par les messages d'erreur et de sollicitation peut alors être calculé et, plus important encore, les fichiers peuvent maintenant être conçus.

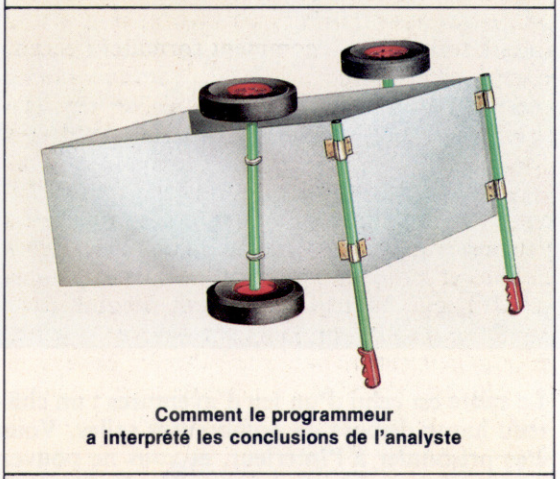
Pour ce programme, où les fichiers renfermeront des blocs de texte de taille importante, répartir le fichier sur plusieurs bandes afin d'effectuer des recherches peut s'avérer être la meilleure solution. Si cela est justifié, les données seront réduites par un algorithme de codage avant d'être écrites sur bande et décodées lors de la lecture.

Les fonctions nécessaires sont maintenant évidentes. Des routines serviront à ajouter et à éditer le texte des données, à stocker le nouveau texte d'entrée (ces routines doivent mettre à jour les index utilisés par le système), à accepter des noms de pièces, à rechercher et à afficher des descriptions, etc. Toutes ces routines doivent être présentées à l'utilisateur comme des options, et elles doivent toutes être en mesure de gérer des données non valables.

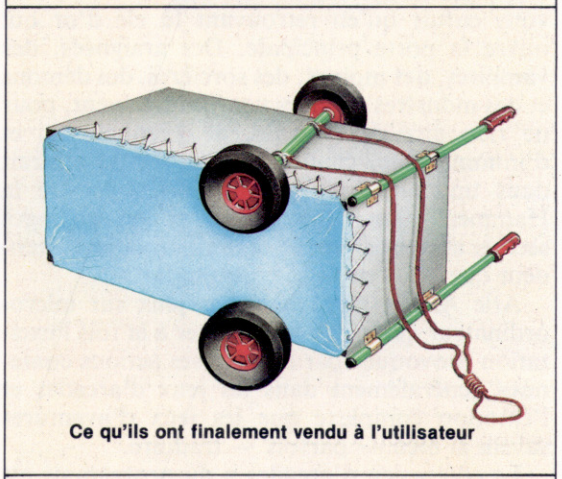
Ici, nous recommandons à l'utilisateur de vérifier soigneusement la conception du programme afin de s'assurer de son bon fonctionnement. Si tout semble parfait, le programme peut maintenant être écrit.



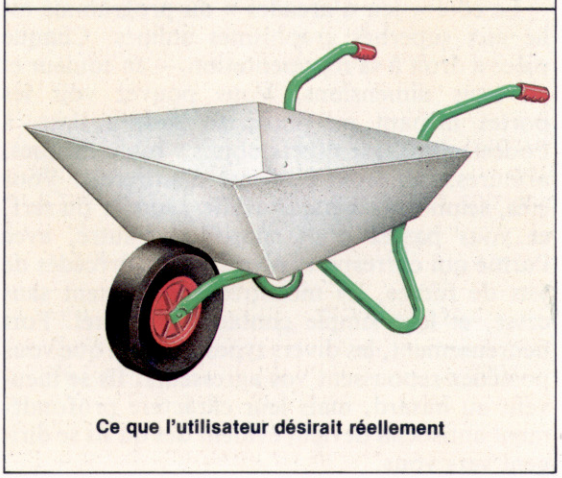
Comment l'analyste percevait les besoins de l'utilisateur



Comment le programmeur a interprété les conclusions de l'analyste



Ce qu'ils ont finalement vendu à l'utilisateur



Ce que l'utilisateur désirait réellement

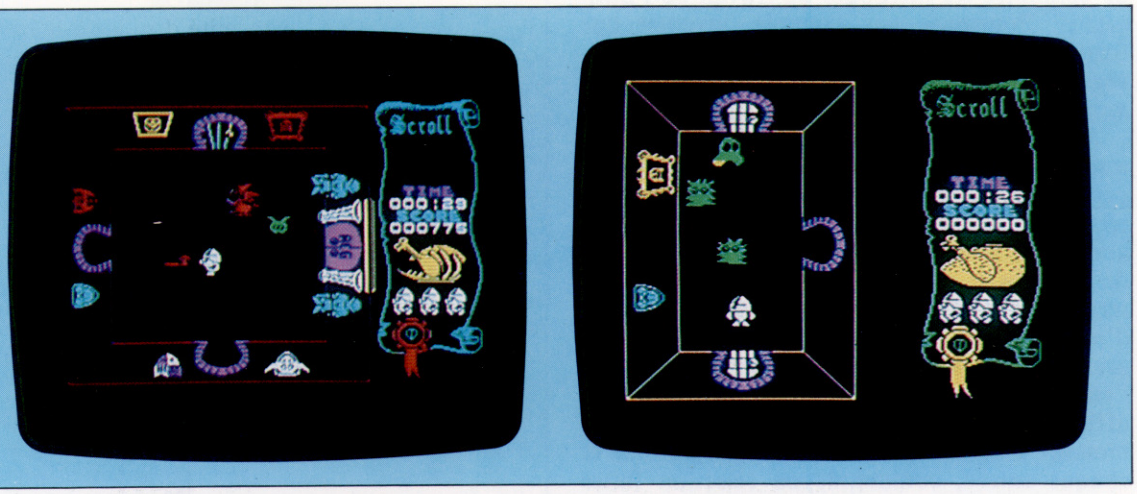
Décrire, définir, concevoir
Si une équipe typique de développement de logiciel, formée de l'utilisateur, de l'analyste et du programmeur tentait de résoudre le problème posé par le transport des charges lourdes dans un potager, voilà ce que cela pourrait produire. Une mauvaise communication — entre experts et non-experts, et entre experts — est toujours le principal problème auquel ont à faire face les équipes de conception.
(Cl. Steve Cross.)

Fantômes à vendre

Ultimate/Play The Game a la réputation de ne produire que des jeux de qualité. Voici Atic Atac, un jeu pour Spectrum qui mêle habilement la rapidité d'action et la réflexion stratégique.

Un jeu risqué

Les reproductions montrent deux moments du jeu qui est à la fois une aventure et un jeu d'arcades très rapide. Tandis que vous courez à travers les passages secrets et les corridors à la recherche d'une clé d'or, les monstres se succèdent à votre poursuite.
(Cl. Liz Heaney.)



Le cadre est celui d'un jeu d'aventures : un château hanté dispose de nombreuses salles. Vous êtes prisonnier à l'intérieur, et vous ne pouvez vous enfuir qu'en retrouvant la clé d'or qui ouvre la porte principale. Des araignées, des vampires, des moines, des sorcières, des démons et des monstres affamés vous poursuivent, pour ne rien dire de Dracula, de Frankenstein et d'innombrables chauves-souris ! On se croirait dans un catalogue de films d'horreur de la Hammer Films. Les pièges et les passages secrets abondent, et de nombreux objets attendent que vous en fassiez usage.

Atic Atac est un des rares jeux sur micro-ordinateur qui arrive à combiner à la fois l'excitation provoquée par les diverses actions contenues généralement dans les jeux d'arcades et l'émotion complexe que les jeux d'aventures savent si bien — parfois — traduire.

Le côté « jeu d'arcades » du programme est lié aux superbes graphismes animés. Chaque pièce a droit à sa représentation — en couleur et en trois dimensions. Vous pouvez voir les portes menant au nord, au sud, à l'est, à l'ouest, ainsi que divers objets : bibliothèques, armures, tableaux, horloges d'autrefois. Vous êtes, selon vos désirs, chevalier, sorcier ou serf, et vous passez d'un endroit à l'autre, avec l'arme qui correspond à votre état. Précédés de jets de fumée, les monstres apparaissent sans cesse, et leur simple contact est mortel. Fort heureusement, les divers types d'armes que vous possédez repoussent vos agresseurs. Ils se meuvent au hasard, mais leur caractère profondément antisocial devient évident dès qu'ils se dirigent vers vous.

Un joystick (de type Kempston) ou bien les touches du clavier permettent de contrôler le jeu. Il est regrettable que, comme bien d'autres jeux, Atic Atac fasse usage des touches Q, W, E, R : elles sont sur la même rangée, l'une à côté de l'autre, ce qui rend les manipulations bien plus difficiles. Il serait bien plus agréable d'utiliser les touches Q et A pour descendre et monter, et certaines des touches (au choix) de la dernière rangée pour aller à gauche et à droite.

Les objets, et surtout la clé d'or, ont un rôle qui permet de sortir du simple cadre « jeu d'arcades ». Certaines portes de couleur ne s'ouvrent que si on a une clé semblable, et les monstres peuvent être tenus à l'écart par telle ou telle de vos possessions. A droite de l'écran, un poulet rôti vous rappelle que vous devez manger, faute de quoi vous mourrez de faim (il se transforme alors en petit tas d'os...).

Atic Atac est d'abord perçu par l'amateur comme un jeu d'arcades très difficile. Ce n'est qu'après avoir maîtrisé certaines des techniques de combat contre les monstres qu'on peut apprécier le côté « jeu d'aventures », en partant à la recherche des objets et du trésor. Mais, soyez prévenu, c'est un jeu que vous devrez pratiquer des heures durant avant de trouver la clé et d'ouvrir la porte...

Atic Atac : Spectrum 48 K.
Auteurs : Ashby Computers and Graphics Ltd.
Éditeurs : Ultimate/Play The Game
Joystick : de type Kempston
Format : cassette

Construction de routine

Les figures graphiques ou « lutins » (sprites) aident à créer des jeux rapides de type arcades en basic. Voici une routine pour créer et déplacer un lutin sur le BBC Micro.

Les lutins peuvent prendre plusieurs formes, mais elles doivent toutes être dessinées sur une grille commune. Il n'y a pas de restriction quant au choix des dimensions de la grille, mais il vaut mieux qu'elle comporte un nombre entier d'octets de largeur (8, 16, 24 bits, etc.). La routine que nous proposons utilise une grille de 24 pixels de largeur sur 21 pixels de profondeur. Seront donc utilisés 63 octets de mémoire pour garder la forme du lutin. Celle-ci est définie par le dessin d'une forme sur la grille, puis par le codage de ce dessin en binaire. Ici, chaque pixel que nous voulons allumer dans la forme définitive est codé par un, et chaque pixel éteint par zéro. Une fois les octets formant le lutin définis en valeurs binaires, ils doivent être convertis soit en décimal, soit en hexadécimal, et placés dans une zone de mémoire.

Les 63 nombres qui définissent le lutin peuvent être entrés par des instructions DATA et lus par READ dans la partie BASIC du programme. Comme chaque nombre est ainsi entré, il doit être placé dans une zone de mémoire spécialement réservée à cette fin. Cette zone peut se trouver n'importe où en RAM, du moment qu'elle est protégée afin qu'on ne puisse pas écrire par-dessus en exécutant le programme. L'emplacement le plus évident pour stocker les données du lutin est le haut de la zone de programme BASIC. Cet emplacement est défini par la variable HIMEM. Pour que nos données ne puissent être écrasées, il est d'abord nécessaire d'abaisser un peu HIMEM. Les lignes 220 et 230 du programme font cela et fixent l'adresse du premier octet de données, SPRDAT, pour commencer juste au-dessus avec la nouvelle valeur de HIMEM. Les lignes 1740 à 1770 lisent les données et les placent dans les 63 octets commençant à l'emplacement SPRDAT.

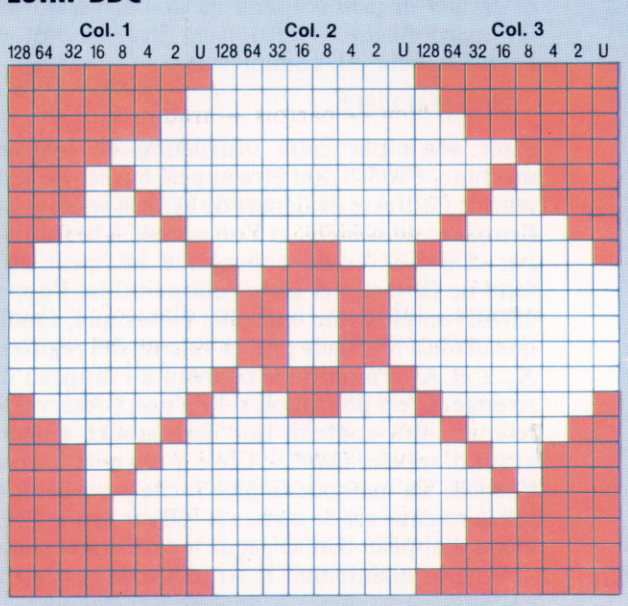
La principale fonction du langage machine est d'analyser le dessin du lutin, puis d'exécuter les opérations nécessaires pour convertir les données en affichage écran. Pour cela, la routine en langage machine doit considérer chaque bit des 63 octets de données, et pour chaque bit décider s'il faut tracer un point (si le bit est un) ou laisser un espace (si c'est zéro). Le moyen le plus aisé d'analyser chaque bit d'un octet particulier est probablement d'utiliser une des instructions de rotation. La ligne 1100 du code source utilise les instructions ROL (rotation à gauche) sur un octet particulier. Cette instruction a pour effet de déplacer chaque bit de l'octet d'une position vers la gauche. L'ancienne valeur du drapeau de retenue est insérée à l'extrémité droite, et le bit qui « tombe » à gauche est décalé dans la retenue. En réitérant ROL, la routine peut examiner chaque bit, lorsqu'il se trouve dans la retenue. A chaque ROL suivant, le bit retourne à l'extrémité droite de l'octet. Tant que nous veillons à ne pas altérer le drapeau de retenue entre les ROL, l'octet retrouvera sa valeur initiale après neuf rotations.

	C								
Contenu initial	C	1	1	0	1	1	1	0	0
– après 1er ROL	1	1	0	1	1	1	0	0	C
– après 2e ROL	1	0	1	1	1	0	0	C	1
– après 3e ROL	0	1	1	1	0	0	C	1	1
– après 4e ROL	1	1	1	0	0	C	1	1	0
– après 5e ROL	1	1	0	0	C	1	1	0	1
– après 6e ROL	1	0	0	C	1	1	0	1	1
– après 7e ROL	0	0	C	1	1	0	1	1	1
– après 8e ROL	0	C	1	1	0	1	1	1	0
– après 9e ROL	C	1	1	0	1	1	1	0	0

Cet exemple montre que le contenu initial du drapeau de retenue (C) ne nous concerne pas particulièrement, puisqu'il est déplacé à travers l'octet pour retourner finalement dans le dra-

Taille du lutin
Le lutin mesure 24 × 31 pixels et occupe ainsi 63 octets (1 pixel par bit en mode haute résolution). La couleur du lutin est initiée par la variable, logcol, et sa taille est déterminée par les 2 facteurs multiplicatifs, XSCALE et YSCALE.
(Cl. Liz Dixon.)

Lutin BBC



Données		
Col. 1	Col. 2	Col. 3
255	0	255
254	0	127
252	0	63
240	0	15
232	0	23
228	0	39
194	0	67
129	24	129
0	189	0
0	102	0
0	102	0
0	102	0
0	189	0
129	24	129
194	0	67
228	0	39
232	0	23
240	0	15
252	0	63
254	0	127
255	0	255

peau de retenue. Notez que le ROL à la ligne 1100 est utilisé en mode d'adressage indexé absolu de sorte que chacun des 63 octets est analysé à son tour de cette manière.

Une fois que la routine a isolé un bit particulier et décidé d'y mettre un point ou un espace, le travail de tracé à l'écran doit être effectué. Il y a deux méthodes pour le BBC Micro. La première consiste à entrer (POKE) directement des valeurs dans la zone de mémoire qui contrôle l'affichage écran. Sur le BBC, ce n'est pas aussi facile qu'il y paraît. Deux problèmes essentiels se posent. Primo, les zones de mémoire écran peuvent varier entre modèles A et B, et entre modes. Secundo, la relation mathématique entre pixels et bits en mémoire écran est très complexe. Par exemple, en mode 2, chaque octet de mémoire ne contrôle que deux pixels sur l'écran. Comme ce mode comporte 16 couleurs, chaque pixel nécessite 4 bits pour définir sa couleur. En mode 0, par contre, mode à 2 couleurs, chaque pixel n'a besoin que d'un bit pour sa définition. Il serait possible d'écrire une routine pour coder notre format de lutin en mode 2, mais cette routine ne marcherait qu'en mode 2.

Heureusement, il existe une autre méthode. En graphiques BASIC, le système d'exploitation du BBC doit faire le tri des manipulations nécessaires. Nous pouvons accéder à cette routine de S.E. pour la plupart des tâches difficiles. De plus, une routine utilisant cet appel au S.E. fonctionnera dans tous les modes graphiques. La routine marche de la même façon qu'une commande VDU en BASIC BBC. Par exemple, pour tracer un seul point à l'écran, la commande VDU suivante pourrait être utilisée :

```
VDU 25,68,300,700;
```

Ici, les coordonnées *x* et *y* sont spécifiées respectivement par les nombres 300 et 700. Cette commande VDU peut être doublée en langage machine en utilisant l'appel au S.E., OSWRCH. Cet appel est réitéré après avoir placé préalablement un nombre dans l'accumulateur. Comme ce dernier ne peut contenir qu'un octet à la fois, les coordonnées *x* et *y* doivent être scindées en octet Lo/octet Hi comme suit :

```
VDU 25,68,44,1,188,2;
```

Pour accomplir cette instruction en langage machine, OSWRCH doit être appelé 6 fois. Le vecteur de l'adresse de départ de OSWRCH se trouve à l'emplacement &FFEE, et l'on accède à la routine par JSR&FFEE. Toute commande VDU peut être faite de la sorte, et cette routine utilise l'appel OSWRCH en plusieurs endroits. Notez que, tandis que OSWRCH n'affecte pas les valeurs des registres X, Y et A, il modifie le contenu du drapeau de retenue. C'est pourquoi, si l'on doit préserver la retenue, son contenu doit être stocké ailleurs avant d'appeler OSWRCH. C'est le cas avec la routine ROL. La méthode la plus facile pour conserver la retenue est de mettre le PSR sur la pile (PHP) avant d'appeler OSWRCH, et de le ressortir ensuite.

Voyons maintenant la structure de la routine en langage machine. L'analyse des données du lutin et du tracé à l'écran se fait principalement en utilisant le sous-programme SPRPLT, qui commence à la ligne 890 du code source. La première opération de cette routine consiste à établir la méthode de tracé comme une opération OU exclusif. C'est l'équivalent de la commande GCOL en BASIC BBC. Les points tracés de cette manière peuvent être effacés en retraçant par-dessus. Toute donnée d'écran en dessous du lutin sera donc laissée intacte. Au début du langage machine, un mouvement absolu est effectué pour positionner le coin supérieur gauche du lutin. Chaque rangée est ensuite analysée en prenant trois octets de données et en leur faisant subir une rotation, comme décrit précédemment.

Un tracé, ou mouvement relatif, est fait sur une distance déterminée par un facteur multiplicatif, XSCALE, dépendant de la valeur du bit de donnée dans la retenue. A la fin de chaque rangée de trois octets, un mouvement absolu est encore effectué, à la même coordonnée *x* en haut et à gauche du lutin, mais à une coordonnée *y* réduite, déterminée par un facteur multiplicatif vertical, YSCALE. Le processus est répété jusqu'à ce que les 63 octets aient été analysés.

Le sous-programme SPRPLT est utilisé en deux endroits. D'abord pour effacer l'ancien lutin en retraçant par-dessus, ensuite, pour tracer le nouveau lutin. Les coordonnées sont alors transférées vers OLDX et OLDY en vue de la prochaine utilisation de la routine.

L'utilisation de la routine en langage machine à partir du basic

La routine peut être utilisée sans difficulté par les programmeurs BASIC, et ne nécessite pas de comprendre la routine elle-même. Les différentes phases en sont :

1. Dessiner le lutin et placer les données dans une zone de mémoire comme indiqué dans le programme.
2. Mettre le mode d'affichage souhaité.
3. Mettre les valeurs de XSCALE et YSCALE comme indiqué à la ligne 1870.
4. Mettre la couleur logique du lutin comme indiqué à la ligne 1890.
5. Mettre les coordonnées *x* et *y* de la position où vous voulez que le lutin apparaisse, et utiliser la procédure donnée aux lignes 2010 à 2060 pour convertir les coordonnées absolues en forme octet Lo/octet Hi.
6. Appeler LUTIN.

La routine en langage machine peut être incorporée dans votre listage basic, comme ici. Le listage assembleur peut être supprimé en remplaçant la ligne 260 par :

```
FOR opt%=0 TO 2 STEP 2
```

Vous pouvez également sauvegarder la routine une fois assemblée (c'est-à-dire après exécution) en utilisant la commande *SAVE, en notant soigneusement les adresses de début et de fin du code, données lorsque le listage d'assemblage est affiché.

Lutin BBC

```
10 REM **** LUTINS BBC ****
20 :
30 REM ** METTRE VARIABLES PAGE ZERO **
40 TYPE    =&70
50 OLDXLO  =&71:?OLDXLO=0
60 OLDXHI  =&72:?OLDXHI=0
70 OLDYLO  =&73:?OLDYLO=0
80 OLDYHI  =&74:?OLDYHI=0
90 NEWXLO  =&75
100 NEWXHI  =&76
110 NEWYLO  =&77
120 NEWYHI  =&78
130 XSCALE  =&79
140 YSCALE  =&7A
150 LO=col  =&7B
160 ROW     =&7C
170 YTMPLO  =&7D
180 YTMPHI  =&7E
190 XTMPLO  =&7F
200 XTMPHI  =&80
210 :
220 HIMEM=HIMEM-150
230 SPRDAT  =HIMEM+1
240 OSWRCH  =&FFEE
250 DIM MC% &01FF
260 FOR opt%=0TO 3 STEP 3
270   P%=MC%
280
290   OPT opt%
300\**** DEPL. VERS ANCIEN X,Y ****
310\
320   SPRITE LDA #25
330          JSR OSWRCH
340          LDA #68
350          JSR OSWRCH
360          LDA OLDXLO
370          STA XTMPLO
380          JSR OSWRCH
390          LDA OLDXHI
400          STA XTMPHI
410          JSR OSWRCH
420          LDA OLDYLO
430          STA YTMPLO
440          JSR OSWRCH
450          LDA OLDYHI
460          STA YTMPHI
470          JSR OSWRCH
480\
490\**** EFFACER ANCIEN LUTIN ****
500\
510          JSR SPRPLOT
520\
530\**** DEPL. VERS NOUVEAU X,Y ****
540  NEWMOV  LDA #25
550          JSR OSWRCH
560          LDA #68
570          JSR OSWRCH
580          LDA NEWXLO
590          STA XTMPLO
600          JSR OSWRCH
610          LDA NEWXHI
620          STA XTMPHI
630          JSR OSWRCH
640          LDA NEWYLO
650          STA YTMPLO
660          JSR OSWRCH
670          LDA NEWYHI
680          STA YTMPHI
690          JSR OSWRCH
700\
710\**** TRACE NOUVEAU LUTIN ****
720\
730          JSR SPRPLOT
740\
750\**** TRANSFERT NOUV. X,Y VERS ANC.X,Y ****
760          LDA NEWXLO
770          STA OLDXLO
780          LDA NEWXHI
790          STA OLDXHI
800          LDA NEWYLO
810          STA OLDYLO
820          LDA NEWYHI
830          STA OLDYHI
840\
850\**** RETOUR AU BASIC ****
860\
870        RTS
880\
890\**** S-P TRACE DE LUTIN ****
900\
910\** TRACE OU EXCLUSIF **
920.SPRPLOT LDA #18
930          JSR OSWRCH
940          LDA #8
950          JSR OSWRCH
960          LDA !o=col
970          JSR OSWRCH
980
990
1000   ** INITIALISE COMPTES **
1010   ** X COMPTE LES OCTETS, Y LES BITS**
1020   ** ROW COMPTE LES RANGS DE 3 OCTETS**
1030         LDX #&00
1040 NEWROW  LDA #&00
1050         STA ROW
1060
1070 BYTE    LDY #&09
1080 BIT     LDA #65
1090         STA TYPE
1100         ROL SPRDAT,X
1110         PHP  STOCKE RETENUE SUR PILE
1120         BCS DOPLOT
1130         LDA #64
1140         STA TYPE
1150\** COMMANDE DE TRACE VDU **
1160 DOPLOT  LDA #25
1170         JSR OSWRCH
1180         LDA TYPE
1190         JSR OSWRCH
1200         LDA XSCALE
1210         JSR OSWRCH
1220         LDA #&00
1230         JSR OSWRCH
1240         JSR OSWRCH
1250         JSR OSWRCH
1260\** FIN DE COMMANDE DE TRACE **
1270         PLP  RECUPERE RETENUE
1280         DEY
1290         BNE BIT
1300\** SI OCTET FINI **
1310         INX
1320         CPX #63
1330         BEQ FINISH
1340\** VERIF. FIN DE RANG **
1350         INC ROW
1360         LDA ROW
1370         CMP #3
1380         BNE BYTE
1390\** SI FIN DE RANG SOUSTRAIRE YSCALE DE Y
1400\
1410         LDA YTMPLO
1420         SEC
1430         SBC YSCALE
1440         STA YTMPLO
1450         BCS NOSUB
1460         DEC YTMPHI
1470\** MOUVT ABSOLU POUR DEBUT NOUV. RANG **
1480\
1490.NOSUB   LDA #25
1500         JSR OSWRCH
1510         LDA #68
1520         JSR OSWRCH
1530         LDA XTMPLO
1540         JSR OSWRCH
1550         LDA XTMPHI
1560         JSR OSWRCH
1570         LDA YTMPLO
1580         JSR OSWRCH
1590         LDA YTMPHI
1600         JSR OSWRCH
1610\
1620\** RANG SUIVANT **
1630         JMP NEWROW
1640\
1650\** FIN DE SOUS-PROGRAMME **
1660.FINISH RTS
1670
1680 )
1690 NEXT
1700 :
1710 REM **** LE PROGR. BASIC COMMENCE ICI ****
1720 :
1730 REM ** LECTURE DES DATA **
1740 FOR address=SPRDAT TO SPRDAT+62
1750 READ DATA:?address=data
1760 NEXT address
1770 :
1780 REM **** METTRE PARAMETRES LANG. MACH. ****
1790 :
1800 MODE1
1810 GCOL0,129
1820 CLG
1830 PXSCALE =4:?YSCALE =4
1870 ?!o=col+1
1880 :
1890 x=700:Y=800
1900 PROCCOOROS (X,Y)
1910 CALL SPRITE
1920 :
1930 REM **** ATTENDRE TOUCHE CURSEUR ****
1940 FOR S=0 TO 1 STEP 0
1950 PROCKEYS
1960 PROCCOORDS (X,Y)
1970 CALL SPRITE
1980 NEXT S
1990 END
2000 :
2010 DEF PROCCOORDS (X,Y)
2020 XH=X DIV 256:XL=X MOD 256
2030 YH=Y DIV 256:YL=Y MOD 256
2040 ?NEWXLO=XL:?NEWXHI=XH
2050 ?NEWYLO=YL:?NEWYHI=YH
2060 ENDPROC
2070 REM **** EXPLORE CLAVIER ****
2080 DEF PROCKEYS
2090 LOCAL LT,ZZ:LST=2
2100 FOR ZZ=0 TO 1 STEP 0
2110 IF INKEY(-58) THEN Y=Y-50:ZZ=LT
2120 IF INKEY(-42) THEN Y=Y-50:ZZ=LT
2130 IF INKEY(-26) THEN X=X-50:ZZ=LT
2140 IF INKEY(-122) THEN X=X+50:ZZ=LT
2150 NEXT ZZ
2160 ENDPROC
2170 REM **** SPRITE DATA ****
2180 DATA 255,0,255,254,0,127,252,0,63
2190 DATA 240,0,15,232,0,23,228,0,39
2200 DATA 194,0,67,129,24,129,0,189,0,
2210 DATA 0,102,0
2220 DATA 0,102,0
2230 DATA 0,102,0
2240 DATA 0,189,0,129,24,129,194,0,67
2250 DATA 228,0,39,232,0,23,240,0,15
2260 DATA 252,0,63,254,0,127,255,0,255
```

Ce que fait le lutin

Le lutin — défini dans les 63 octets de DATA — peut être déplacé à travers l'écran par les touches fléchées. Sa lenteur relative est la conséquence de l'utilisation de la routine ROM OSWRCH, mais il marche dans tous les modes. Lorsque le programme est exécuté (RUN), le listage d'assemblage remplit d'abord l'écran, puis l'affichage graphique se produit.

LLAMA SOFT

Llamasoft est une compagnie qui repose sur le seul talent de Jeff Minter, un programmeur devenu très vite célèbre grâce à des jeux d'un humour violemment absurde tournant sur les ordinateurs Commodore et Atari.

La plupart des firmes productrices de logiciels sont créées par un ou deux programmeurs qui travaillent à domicile. Une fois le succès venu, ils embauchent du personnel nouveau, et le « style maison », qui a fait leur réussite disparaît peu à peu, parce que la nouvelle compagnie doit assurer une production incessante. Ce n'est pourtant pas le cas de Llamasoft : personne d'autre n'aurait pu sortir des jeux comme Revenge of The Mutant Camels ou Sheep in Space.

Ce style si particulier est celui d'un seul homme. Jeff Minter n'a que vingt-deux ans, mais il est devenu en trois ans une des stars des jeux vidéo. L'image de Llamasoft doit tout à son obsession des lamas et des chameaux, et son adjectif favori, « terrifiant » (awesome !), est le maître mot de la publicité de la firme. Minter a même — consécration suprême — été brocardé par d'autres programmeurs : l'une des séquences du Manic Miner de Matthew Smith s'appelle par exemple « Attack of the Mutant Telephones ».

Minter écrivit ses premiers jeux en 1981, après avoir quitté l'université d'East Anglia où il étudiait les maths et la physique. Il était premier en informatique, mais rata l'examen de mathématiques et fut contraint d'abandonner.

Revenge of The Mutant Camels
Cette image de Revenge of The Mutant Camels (la suite de Attack of The Mutant Camels) illustre bien le style volontiers surréaliste de Minter — qu'on a aussi comparé à Jean-Luc Godard !

« Your 64 and Vic-20 ».

En 1982, il écrivit pour le Vic-20 une version de Defender et créa Llamasoft pour le commercialiser. Le succès fut assuré dès la parution de Traxx et Gridrunner ; ce dernier se vendit remarquablement en Angleterre et aux États-Unis. La mère de Minter, Hazel, qui s'occupe de la gestion, souligne : « Gridrunner a été numéro un aux États-Unis pendant un temps fou. C'est vraiment grâce à lui que Jeff s'est fait connaître. »

Les jeux de Minter sont pour l'essentiel des histoires de combats galactiques ; mais à chaque fois, ce thème rebattu se voit renouvelé par des trouvailles volontiers absurdes. Dans Revenge of the Mutant Camels, les chameaux mutants, de 30 m de hauteur et protégés par des boucliers au neutronium, affrontent ainsi des kangourous skieurs, des œufs sur le plat, des cabines téléphoniques et... des ordinateurs Spectrum. Minter voit en effet dans ce dernier « une machine atroce », et refuse d'écrire pour lui, bien que Salamander Software et Quicksilva aient récemment lancé des versions de ses jeux destinées à cet appareil. Llamasoft compte bientôt embaucher un nouvel assistant, qui fera le même travail sous la supervision de Minter : « Ce qui m'intéresse, c'est d'écrire un bon jeu. Ce sont mes idées à moi, et il n'y a pas de raison que d'autres personnes le fassent. »

Llamasoft, qui est devenue une S.A.R.L. en avril 1984, est une entreprise très familiale, dirigée depuis la maison des Minter à Tadley, Hampshire. Patrick Minter, le père, aide son fils dans le travail de programmation, la mère s'occupe de la gestion. La firme emploie deux assistants, deux employés et un graphiste.

Minter a déjà écrit plus d'une vingtaine de jeux, destinés aux ordinateurs Atari et Commodore. Outre une grande maîtrise du graphisme, ils se caractérisent par le retour fréquent de thèmes animaliers — d'abord les lamas, puis les chameaux (« Je suis en plein dans les chameaux »). Les premiers camélidés firent leur apparition en 1982 dans Attack of the Mutant Camels (pour Atari), et réapparaissent dans Revenge of the Mutant Camels, l'un des plus beaux jeux jamais écrits pour le Commodore 64. Il y a aussi des chèvres, des moutons (Sheep in Space)...

En dépit de son très gros succès, Minter ne compte pas changer de type de gestion. L'aspect commercial ne l'intéresse guère, et il continue à donner libre cours à son obsession des animaux à fourrure en écrivant de nouveaux jeux « terrifiants ».

PROGRAMME N° 26

TRACÉ D'UN HISTOGRAMME

Derrière ce nom un peu barbare se cache un graphique tout simple, visualisé, qui est fonction des valeurs introduites (comme par exemple le chiffre d'affaires annuel sur plusieurs années d'une P.M.E.) et d'un nombre proportionnel, de symbole $ ou *, s'inscrivant horizontalement sur l'écran.

Une fois toutes les valeurs introduites, l'écran offre une représentation graphique de ces données et permet des interprétations très intéressantes (ici l'évolution conjoncturelle d'un chiffre).

Voici le programme et ses lignes de commentaires en REM :

```
10 REM tracé d'un histogramme
20 REM VA       :tableau des valeurs
30 REM V1       :VAL MINI
40 REM V2       :VAL MAXI
50 REM N        :NB DE VALEURS
60 REM S$       :caractère de visualisation utilisé
61 REM $ ou * ou tout autre
70 REM C        :nom du caractère d'une valeur
80 REM R$       :chaine de caractères d'une valeur
90 REM on dimensionne la table
100 DIM VA (400)
105 ?:?"NOMBRE DE VALEURS"
```

N nombre de valeurs (N ne doit pas dépasser 400). Si N < 0 alors arrêt du programme.

```
110 INPUT N     :IF N  = 0 THEN GOTO 370
```

On entre chacune de *n* valeurs dans la variable VA (I). I représente le rang de la valeur entrée. I varie de 1 à N.

```
120 FOR I = 1 TO N
130 INPUT "VALEUR:";VA(I)
140 NEXT I
```

V1 valeur mini entrée, V2 valeur maxi entrée. Au début V1 = 0 et V2 est supposée égale à la première valeur entrée : ici VA (I).

```
150 V1 = 0      :V2 = VA (1)
```

Pour chacune des valeurs du rang 2 au rang N on effectue 2 tests :
si la valeur est plus petite que le minimum connu, alors on réactualise ce minimum. De même, si cette valeur est plus grande que le maximum connu, on réactualise ce maximum. Arrivé au bout de la liste des valeurs, on connaît les deux valeurs minimales et maximales.

```
156 FOR I = 2 TO N
160 IF VA (I)  V1  THEN V1 = VA (I)
170 IF VA (I)  V2  THEN V2 = VA (I)
180 NEXT I
```

S$ caractère de visualisation permettant la représentation de l'histogramme (par exemple on tapera * et ainsi cet histogramme sera construit avec des étoiles).

```
190 INPUT "CARACTERE DE VISUALISATION";S$
```

N est le nombre de caractères sur lesquels s'étend l'histogramme. Ne pas oublier qu'une ligne d'écran comporte 40 ou 80 caractères. La valeur minimale est représentée par un seul caractère. La valeur maximale par NB caractères.

```
200 INPUT "NOMBRE DE CARACTERES MAX";NB
```

E nombre d'unités représentées par un caractère.

```
205 E=(V2 - V1 ) / (NB - 1 )
```

Affichage des valeurs minimales et maximales.

```
210 ? TAB ( 1 );" N ";TAB (4) ; V1 ; TAB ( NB); V
```

Pour chacune des valeurs entrées (I représente le rang de la valeur; I varie donc de 1 à N) on calcule C, le nombre de caractères représentant la valeur entière VA (I).

```
220 FOR I = 1 TO N
230 C     = 1 + INT (( VA (I) - V1) / E
240 R$    =""
```

R$ chaîne de caractères de longueur C constituée d'une succession de caractères S$ donnés précédemment.

```
250 FOR J= 1 TO C
260 R$    = R$ + S$
270 NEXT J
```

Affichage de rang I de la valeur et de la chaîne de caractères R$.

```
280 ? TAB ( 1) ; I ; TAB ( 4) ; VA (I) ; TAB (10) ; R$
```

Affichage du rang I de la valeur de la chaîne de caractères R$.

```
290 NEXT I
```

On passe à la valeur suivante.

```
300 ?:?"un caractère vaut:";E
```

Affichage du nombre d'unités. E X $: on entre OUI si on veut choisir un nouveau caractère de visualisation, changer le nombre de caractères sur une ligne (retour en 190). On entre NON si l'on veut rentrer une nouvelle série de données.

```
310  ?"VOULEZ-VOUS UNE NOUVELLE"
320  ?"VISUALISATION"(OUI/NON)";X $
330 INPUT X $
340 IF X $  ="OUI"GOTO 190
350 IF X $  ="NON"GOTO 105
```

Si X $ n'est ni OUI ni NON, retour à sa lecture en 310.

```
360 GOTO 310
370 FIN
```

Après avoir introduit ce programme et tapé RUN, la machine affiche : NOMBRE DE VALEURS?

On tape 10, si on veut introduire 10 valeurs, on tapera 24 pour 24 valeurs.

Ensuite, on vous demande d'entrer les valeurs l'une après l'autre; vous les tapez à côté du message

"VOTRE VALEUR?"

Le programme vous demande à la fin de la saisie : CARACTÈRE DE VISUALISATION; vous taperez + si vous désirez que les caractères soient des + par exemple.

De même, à la question NOMBRE DE CARACTÈRES? vous taperez 20, si vous désirez que l'histogramme s'étende sur 20 caractères.

Enfin, le programme affiche l'histogramme des valeurs avec leur rang.